

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

# МОДЕЛЮВАННЯ ФІЗИЧНИХ ПРОЦЕСІВ

## комп'ютерний практикум

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського  
як навчальний посібник для студентів,  
які навчаються за спеціальністю 105 «Прикладна фізика та наноматеріали»,  
спеціалізацією «Прикладна фізика»*

Київ  
КПІ ім. Ігоря Сікорського  
2021

Моделювання фізичних процесів: Комп'ютерний практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 105 «Прикладна фізика та наноматеріали», спеціалізації «Прикладна фізика» / КПІ ім. Ігоря Сікорського ; уклад.: Н. Ф. Димитрієва. – Електронні текстові дані (1 файл: 11,6 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 96 с.

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 7 від 13.05.2021 р.)  
за поданням Вченої ради Фізико-технічного інституту (протокол № 7 від 01.03.2021 р.)*

Електронне мережне видання

# МОДЕЛЮВАННЯ ФІЗИЧНИХ ПРОЦЕСІВ КОМП'ЮТЕРНИЙ ПРАКТИКУМ

Укладач *Димитрієва Наталія Федорівна*, канд. фіз.-мат. наук

Відповідальний  
за випуск *Литвинова Т. В.*, канд. техн. наук, доц.

Рецензенти: *Кондаков В. О.*, канд. фіз.-мат. наук

Навчальний посібник призначено для студентів, які навчаються за спеціальністю 105 «Прикладна фізика та наноматеріали». Рекомендації навчають структурувати й формалізувати в термінах модельного представлення широкого класу задач гідрогазодинаміки, процесів тепло- та масообміну та їхньої чисельної реалізації у відкритих пакетах прикладних програм.

© КПІ ім. Ігоря Сікорського, 2021

## ЗМІСТ

Передмова .....	4
Рекомендоване програмне забезпечення .....	5
Кредитний модуль «Моделювання фізичних процесів – 1 частина».	
Етапи моделювання .....	7
1.1 Розрахунок нестационарної ламінарної течії в каверні .....	8
1.2 Розрахунок стаціонарної течії в каналі з уступом .....	21
1.3 Будування неструктурованої розрахункової сітки .....	28
1.4 Будування блочно-структурованої розрахункової сітки .....	40
1.5 Обробка та візуалізація результатів розрахунку в графічних пакетах .....	55
1.6 Особливості використання динамічних сіток .....	66
Кредитний модуль «Моделювання фізичних процесів – 2 частина».	
Задачі фізики суцільних середовищ .....	70
2.1 Порівняння моделей турбулентності .....	71
2.2 Розрахунок процесів теплообміну в кімнаті з підігрівом .....	74
2.3 Розрахунок течії двофазного середовища .....	78
2.4 Розрахунок течії стисливого газу .....	84
2.4.1 Дозвукова течія. ....	85
2.4.2 Надзвукова течія. ....	87
2.5 Розрахунок течій в каналах з заданим градієнтом (тепловий, дифузійний або МГД) .....	91
Рекомендована література .....	94

## ПЕРЕДМОВА

Дані методичні вказівки складено відповідно до програми навчальної дисципліни «Моделювання фізичних процесів» для студентів, які навчаються за спеціальністю 105 «Прикладна фізика та наноматеріали».

Відповідно до навчального плану дисципліна складається з двох кредитних модулів, що включають наступні базові структурні елементи: сучасний стан і можливості комп'ютерного моделювання, методи та етапи моделювання, особливості моделювання нестационарних та стаціонарних задач стисливого та нестисливого середовища при ламінарному та турбулентному режимах, а також процесів тепломасопереносу. Теми та порядок виконання робіт комп'ютерного практикуму сформовано в логічній послідовності і повністю узгоджуються з лекційним матеріалом. Для самостійної роботи студентів до кожної роботи наводиться список рекомендованої літератури.

Метою дисципліни є формування у студентів практичних навичок з основ застосування сучасних комп'ютерних обчислювальних технологій у сфері прикладної діяльності, що пов'язана з фізикою суцільного середовища і енергетичних систем в широкому класі режимів процесів та геометричних конструкцій об'єктів, що вивчаються. У процесі виконання робіт комп'ютерного практикуму студент повинен продемонструвати вміння застосовувати на практиці теоретичні знання, здобуті під час прослуховування курсу лекцій та самостійного вивчення дисципліни.



## РЕКОМЕНДОВАНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

Рекомендовано використання вільно розповсюджуваного програмного продукту, наприклад, відкритих пакетів прикладних програм SALOME [1], OpenFOAM [2], Paraview [3].

Одним з найбільш перспективних відкритих пакетів, що швидко розвиваються, є OpenFOAM. Він розробляється світовою OpenFOAM спільнотою під загальною координацією OpenCFD Ltd і знаходиться у відкритому легальному доступі, який не передбачає ліцензування. В основі його вихідного коду лежить набір бібліотек, що надають інструменти для вирішення ряду прикладних задач, а також проводити розпаралелювання розрахунків в суперкомп'ютерних системах. Пакет працює під різними версіями відкритої ОС Linux (CentOS, Ubuntu, Fedora, Debian та інші).

Чисельне моделювання системи рівнянь руху в пакеті OpenFOAM проводиться методом кінцевих об'ємів в декартовій системі координат. Інтеграли по контрольному об'єму зводяться до поверхневого методом Гаусса, а значення функції на поверхні комірки інтерполюється із значень функції в центроїдах сусідніх комірок. Робочою мовою коду є об'єктно-орієнтована мова програмування C++, в термінах якого більшість математичних диференціальних і тензорних операторів в програмному коді рівнянь може бути представлена в зрозумілій формі, а метод дискретизації та розв'язку для кожного оператора може бути обраний вже користувачем в процесі розрахунку.

В результаті багаторічного академічного та промислового розвитку на сьогоднішній момент пакет OpenFOAM дозволяє розв'язувати задачі гідродинаміки ньютонівських і неньютонівських в'язких рідин як в нестисливому, так і стисливому наближенні з урахуванням конвективного теплообміну та дією сил гравітації з використанням різних моделей турбулентності, можливо розв'язання дозвукових і надзвукових задач; багатофазні задачі, в тому числі з описанням хімічних реакцій компонент потоку; задачі, що пов'язані з деформацією розрахункової сітки; зв'язані задачі; розпаралелювання розрахунків як в кластерних, так і багатопроцесорних системах.

Візуалізацію результатів розрахунків, проведених із застосуванням відкритого пакету OpenFOAM, пропонується виконувати з використанням відкритого графічного інтерфейсу ParaView. Пакет було розроблено з метою аналізу дуже великих об'ємів інформації з використанням розподілених обчислювальних ресурсів пам'яті і може працювати як на кластерах для аналізу

даних тера-порядку, так і на персональних комп'ютерах для порівняно менших об'ємів.

Дискретне представлення обчислювальної області (сітка) є складовою частиною чисельного моделювання і значною мірою визначає успіх рішення задачі. У вільно розповсюджуваному програмному пакеті прикладних програм OpenFOAM існують різні рівні опису розрахункової сітки. Основний клас polyMesh будується з використанням мінімального об'єму інформації, що необхідна для визначення геометрії сітки. Підтримуються як структуровані багатоблокові сітки, так і не структуровані, в тому числі з багатогранными осередками (polyhedral). Утиліта blockMesh поставляється разом з пакетом OpenFOAM призначена для генерації гексаедральних структурованих сіток. У файлі blockMeshDict задаються елементи і параметри розбиття: вершини, ребра, блоки, зовнішні границі.

Перспективним і функціональним сітковим генератором, особливо для тіл зі складною геометрією, вважається відкрита інтегрована платформа SALOME. Вона дозволяє створювати і редагувати, імпортувати і експортувати CAD моделі, накладати на них сітку за допомогою різних алгоритмів, пов'язувати фізичні параметри і геометрію. Однією з переваг SALOME є можливість перетворення отриманої розрахункової сітки до форматів, які використовуються в різних комерційних і відкритих пакетах прикладних програм. Створена за допомогою Salome сітка і збережена у форматі UNV, конвертується в формат OpenFOAM командою `ideasUnvToFoam mesh.unv`, де `mesh.unv` - ім'я файлу.

**КРЕДИТНИЙ МОДУЛЬ**  
**«МОДЕЛЮВАННЯ ФІЗИЧНИХ ПРОЦЕСІВ – 1 ЧАСТИНА»**  
**ЕТАПИ МОДЕЛЮВАННЯ**

Кредитний модуль «Моделювання фізичних процесів – 1 частина» знайомить студентів з інтерфейсом і основними можливостями відкритих пакетів SALOME, OpenFOAM, Paraview. В ході виконання робіт студенти отримають навички комп'ютерного моделювання типових задач течій нестисливої рідини за всіма етапами: підготовка вихідних даних, побудова сітки, обчислення просторово-часових полів фізичних величин, контроль збіжності, обробка та візуалізація результатів розрахунків, тощо.

Робочою програмою даного кредитного модуля передбачено виконання робіт комп'ютерного практикуму в запропонованій логічній послідовності. Теми робіт взаємозв'язані, повністю відповідають теоретичному матеріалу лекційних занять.

## Робота 1.1. Розрахунок нестационарної ламінарної течії в каверні

Лекційний матеріал. Розділ 3-7

Література: [4-14]

### 1.1.1 Постановка задачі

Задача: cd \$FOAM\_RUN/tutorials/incompressible/icoFoam/cavity/cavity

Посилання: <https://www.openfoam.com/documentation/tutorial-guide/tutorialse2.php#x6-60002.1>

Досліджується нестационарна двовимірна ізотермічна ламінарна течія нестисливої однорідної ньютонівської рідини в каверні з рухомою верхньою стінкою (рис. 1.1.1).

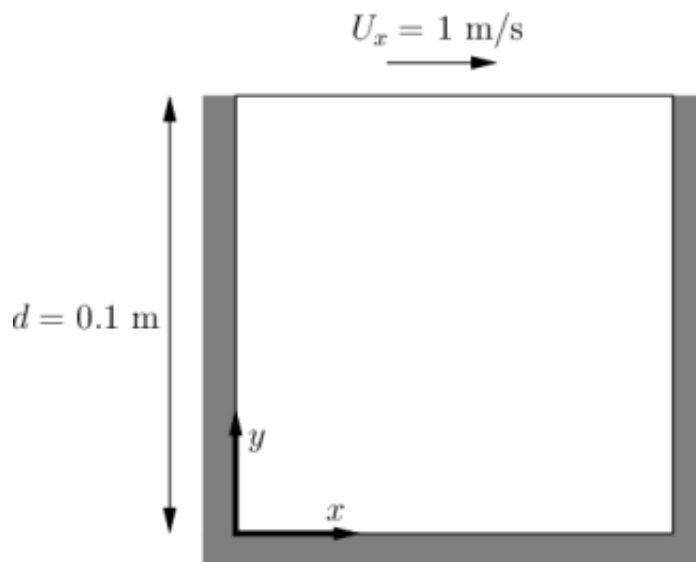


Рис. 1.1.1 Геометрія задачі

Система визначальних рівнянь складається з рівняння Нав'є-Стокса нерозривності і записується наступним чином:

$$\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U} \cdot \nabla) \mathbf{U} + \nu \Delta \mathbf{U} = -\nabla \frac{p}{\rho}, \quad (1)$$

$$\nabla \cdot \mathbf{U} = 0, \quad (2)$$

де  $\mathbf{U} = \{U_x, U_y, U_z\}$  – вектор швидкості,  $p$  – тиск,  $\rho$  – густина рідини,  $\nu$  – коефіцієнт кінематичної в'язкості.

### 1.1.2 Підготовка вихідних даних (Preprocessing)

Будь яка задача, що розв'язується в пакеті OpenFOAM обов'язково містить наступні папки:

- **0** – початкові та граничні умови для кожної змінної фізичної величини;
- **constant** – розрахункову сітку і фізичні властивості середовища;
- **system** – параметри дискретизації і інтегрування рівнянь, .

Для коректних розрахунків важливо:

- правильно задати розмірність;
- коректно задати граничні умови;
- перевірити якість сітки;
- вміти аналізувати процес збіжності.

#### 1.1.2.1 Розрахункова сітка

Пакет OpenFOAM завжди працює в тривимірній декартовій системі координат. Для 2D задач в третьому вимірі задається одна комірка та «пусті» граничні умови empty, що виключають розрахунок в даному напрямку.

Проста геометрія, що складається з квадрата стороною  $d = 0.1$  м в площині x-y, дозволяє побудувати структуровану (регулярну) розрахункову сітку. Структуру блоку показано на рис. 1.1.2. Для цього використовуємо вбудований в пакет OpenFOAM генератор сіток **blockMesh**.

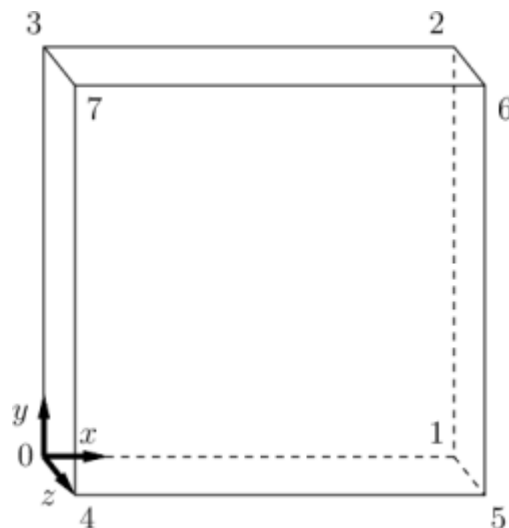


Рис. 1.1.2 Структура блоку

Утіліта `blockMesh` будує сітки з опису, зазначеного у файлі `blockMeshDict`, що знаходиться в системному каталозі даної задачі. Для визначення сітки потрібна наступна інформація:

- **ConvertToMeters** (в старих версіях OpenFOAM) або **scale** (в нових) – параметр, що показує масштаб (необов'язковий). OpenFOAM працює з розмірними величинами. Якщо масштаб не вказано, сітка автоматично будується в метрах.

```
scale 0.1; // означає, що координати точок сітки вказано в дециметрах
```

- В розділі **vertices** задаються координати точок для побудови геометрії:

```
vertices
(
  (0 0 0)
  (1 0 0)
  (1 1 0)
  (0 1 0)
  (0 0 0.1)
  (1 0 0.1)
  (1 1 0.1)
  (0 1 0.1)
);
```

- В розділі **blocks** вказується тип блоку, його вершини і параметри розбиття блоку:

```
blocks
(
  hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1)
```

- 1) Перша група чисел – номери точок у вершинах блоку. Порядок, в якому вказуються точки блоку, визначають яка буде система координат (ліва або права). У всіх блоків порядок повинен бути однаковий, інакше можуть з'являтися блоки з від'ємним об'ємом.
- 2) Друга група показує число розбиття вздовж відповідних вісей (x y z).
- 3) Третя група визначає коефіцієнт зростання комірок в напрямку відповідних вісей. Коефіцієнт дорівнює відношенню довжин останньої і першої комірок.

- В розділі **edges** задаються криволінійні ребра (дуги кіл або сплайни). В даній задачі – пустий блок.
- В розділі **boundary** визначаються поверхні для завдання граничних умов:

```

boundary
(
  movingWall
  {
    type wall;
    faces
    (
      (3 7 6 2)
    );
  }
  fixedWalls
  {
    type wall;
    faces
    (
      (0 4 7 3)
      (2 6 5 1)
      (1 5 4 0)
    );
  }
  frontAndBack
  {
    type empty;
    faces
    (
      (0 3 2 1)
      (4 5 6 7)
    );
  }
);

```

Групи поверхонь задаються списком із граней, кожна з яких визначається чотирма точками. Також необхідно вказати тип граничної умови (Таблиця 1).

## Класифікація поверхонь

Тип поверхні	Опис
patch	загальний тип, що не містить геометричної або топологічної інформації про сітку, наприклад, використовується для входу або виходу потоку.
wall	тверда стінка – поверхня, де можуть використовуватися пристінкові функції (часто в моделях турбулентності)
symmetryPlane	площина симетрії
empty	2d плоска задача, для цієї поверхні виключаються обчислення
wedge	2d вісесиметрична задача, для цієї поверхні виключаються обчислення
cyclic	Поверхня, що циклічно повторюється

Рекомендується перевірка сітки на наявність помилок утілюю **checkMesh**. Висновок програми checkMesh включає в себе наступні розділи:

- загальна статистика (кількість елементів сітки розмірності від 0D до 3D):

Mesh stats

```

points:                882
  internal points:      0
  faces:                1640
  internal faces:       1760
  cells:                400
  boundary patches:     3
  point zones:          0
  face zones:           0
  cell zones:           0

```

- статистика типів комірок:

Overall number of cells of each type:

```

hexahedra:    400
prisms:       0
wedges:       0
pyramids:     0
tet wedges:   0
tetrahedra:   0
polyhedra:    0

```



- загальна топологія сітки:

Checking topology...

Boundary definition OK.  
Cell to face addressing OK.  
Point usage OK.  
Upper triangular ordering OK.  
Face vertices OK.  
Number of regions: 1 (OK).

- топологія зовнішніх границь;

Checking patch topology for multiply connected surfaces ...

Patch	Faces	Points	Surface topology
movingWall	20	42	ok (non-closed singly connected)
fixedWalls	60	122	ok (non-closed singly connected)
frontAndBack	800	882	ok (non-closed singly connected)

- геометричні характеристики комірок (неортогональність, скошеність, масштабність і ін.):

Checking geometry...

Overall domain bounding box (0 0 0) (0.1 0.1 0.01) // мінімальне поле, що містить сітку

Mesh (non-empty, non-wedge) directions (1 1 0) // означає, що x та y не пусті напрямки, а z – пустий для плоскої задачі або клин для вісесиметричної.

Mesh (non-empty) directions (1 1 0)

All edges aligned with or perpendicular to non-empty directions.

Boundary openness (8.47033e-18 -8.47033e-18 -5.901e-17) OK.

Max cell openness = 1.35525e-16 OK.

Max aspect ratio = 1 OK. // Примітка: співвідношення сторін 1 є найкращим.

Minimum face area = 2.5e-05. Maximum face area = 5e-05. Face area magnitudes OK.

Min volume = 2.5e-07. Max volume = 2.5e-07. Total volume = 0.0001. Cell volumes OK.  
// різниця між мінімальним та максимальним об'ємами повинна бути якомога меншою або їх зміна повинна бути максимально плавною

Mesh non-orthogonality Max: 0 average: 0 // вимірюється кут між лінією, що з'єднує два центроїда сусідніх комірок, і нормаль їхньої спільної поверхні. Значення параметру неортогональності 0 є найкращим

Non-orthogonality check OK.

Face pyramids OK.

Max skewness = 1e-08 OK. // вимірюється відстань між перетином лінії, що з'єднує два центроїда сусідніх комірок, і центром їхньої спільної поверхні – чим менше, тим краще

Coupled point location match (average 0) OK.

Якщо побудована сітка відповідає всім критеріям, що висуває пакет OpenFoam для чисельного розрахунку поставленої задачі, вкінці з'явиться:

Mesh OK.

### 1.1.2.2 Крайові умови

Початкові та граничні умови задаються в каталозі 0 для кожної змінної фізичної величини. Для даної задачі ламінарної течії нестисливої рідини змінними являються тиск  $p$  та вектор швидкості  $U$ .

В якості прикладу розглянемо структуру файлу U з крайовими умовами:

- розмірність фізичної величини:

```
dimensions [0 1 -1 0 0 0 0];
```

Цифри в квадратних дужках означають степені основних фізичних величин згідно з Таблицею 2. Таким чином, розмірність швидкості –  $m^1s^{-1}$ .

Таблиця 2

Розмірності фізичних величин

№	Характеристика	Одиниця виміру	Символ
1	Маса	кілограм	кг
2	Довжина	метр	м
3	Час	секунда	с
4	Температура	Кельвін	К
5	Кількість речовини	моль	моль
6	Сила струму	Ампер	А
7	Сила світла	кандела	Кд

**Зверніть увагу!** Розмірність тиску в нестисливій рідині задано як  $m^2/s^2$ . У зв'язку з тим, що густина постійна  $\rho = \text{const}$ , пакет OpenFOAM використовує в розрахунках кінематичний тиск  $p/\rho$ .

- початкові умови:

```
internalField uniform (0 0 0); // в початковий момент задано постійне значення  
швидкості  $U_x = 0$ ;  $U_y = 0$ ;  $U_z = 0$ 
```

- граничні умови:

```
boundaryField
{
    movingWall    // верхня стінка рухається з горизонтальною швидкістю  $U_x = 1$  м/с
    {
        type      fixedValue;
        value      uniform (1 0 0);
    }
    fixedWalls    // на інших стінках задано умову прилипання
    {
        type      noSlip;
    }
    frontAndBack // на передній і задній стінках задано пусті граничні умови, щоб
                  виключити обчислення в напрямку z.
    {
        type      empty;
    }
}
```

### 1.1.2.3 Фізичні властивості середовища

Дана задача є нестисливою і ламінарною, тому достатньо вказати тільки одну величину, яка визначає фізичні властивості середовища – коефіцієнт кінематичної в'язкості  $\nu$ . Ця величина задається в файлі `constant/transportProperties`:

```
nu [0 2 -1 0 0 0] 1e-06; // означає  $\nu = 10^{-6}$  м2/с
```

### 1.1.3 Чисельна модель

Використовується чисельна модель (solver) **icoFoam**, програмні коди якої можна знайти в каталозі [openfoam /applications /solvers /incompressible /icoFoam](http://openfoam.com/applications/solvers/incompressible/icoFoam)

#### 1.1.3.1 Налаштування чисельних схем

У відкритому пакеті прикладних програм OpenFOAM інформацію про дискретизацію системи визначальних рівнянь (1, 2) можна знайти у файлі `system/fvSchemes`:

```
ddtSchemes    // перша похідна за часом
{
    default     Euler;
}
```

```

gradSchemes          // градієнт
{
    default      Gauss linear;
    grad(p)      Gauss linear;
}
divSchemes            // дивергенція
{
    default      none;
    div(phi,U)   Gauss linear;
}
laplacianSchemes      // оператор Лапласа
{
    default      Gauss linear orthogonal;
}
interpolationSchemes  // інтерполяція на грані
{
    default      linear;
}
snGradSchemes         // похідна по нормалі
{
    default      orthogonal;
}

```

Наведено схеми дискретизації для кожної змінної фізичної величини похідних за часом та простором (дивергенція, градієнт, оператор Лапласа), порядок точності, схеми інтерполяції на грані та ін. За часом використано стандартну явну схему дискретизації Ейлера першого порядку точності. Для градієнту тиску і дивергенції швидкості – стандартну дискретизацію скінченного об'єму гауссівського інтегрування, що вимагає інтерполяції значень від центрів комірок до центрів граней.

### 1.1.3.2 Налаштування методів розв'язку СЛАР

У файлі розрахункової задачі `system/fvSolutions` описуються методи розв'язку системи лінійних алгебраїчних рівнянь (СЛАР), значення абсолютної і відносної похибки:

```

solvers
{
    p
    {
        solver      PCG;
        preconditioner DIC;
        tolerance    1e-06;
        relTol       0.05;
    }
}

```

Для розв'язку отриманої системи лінійних алгебраїчних рівнянь застосовувалися ітераційні методи спряжених градієнтів PCG для симетричних матриць. В якості передобумовника була обрана процедура DIC, що заснована на спрощеній схемі неповної факторизації Холецького.

```
U
{
    solver      smoothSolver;
    smoother    symGaussSeidel;
    tolerance    1e-05;
    relTol      0;
}
```

Ітераційний розв'язувач smoothSolver для симетричних та асиметричних матриць, який використовує більш плавний час, вибраний під час прогону, наприклад, метод Гаусса-Зейделя, щоб розв'язок зійшовся до необхідної точності.

Також файл fvSolutions містить опис процедури сумісної корекції полів швидкості та тиску:

```
PISO
{
    nCorrectors    2;
    nNonOrthogonalCorrectors 0;
    pRefCell      0;
    pRefValue     0;
}
```

PISO (Pressure Implicit with Splitting of Operator) – неявний метод зв'язку полів тиску і швидкості, що призначений для розв'язку нестационарних задач, вимагає більше 1 коректора, але зазвичай не більше 4. Згідно з процедурою PISO, формуються три матриці ( $U_x$ ,  $U_y$ ,  $U_z$ ) рівняння збереження імпульсу. Системи розв'язуються послідовно з використанням тиску з попереднього кроку.

### 1.1.3.3 Розрахунковий експеримент

Контроль за ходом розрахунків здійснюється в файлі system/controlDict. У цьому файлі прописуються:

```

application  icoFoam;    // чисельна модель (solver)
startFrom    startTime;  // початковий момент часу
startTime    0;
stopAt       endTime;    // кінцевий момент часу
endTime      0.5;
deltaT       0.005;      // крок за часом
writeControl  timeStep;
writeInterval 20;        // крок виведення результатів
writeFormat  ascii;      // формат виведення результатів
writePrecision 6;        // точність виведення результатів

```

Під час обчислень важливо слідкувати за виконанням умови Куранта:

$$Co = \frac{\delta t |U|}{\delta x} < 1 \quad (3)$$

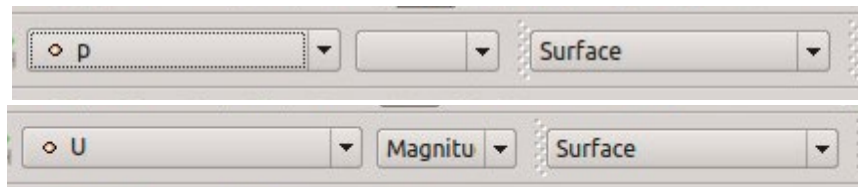
де  $Co$  – число Куранта,  $\delta x$  – розмір комірки,  $\delta t$  – крок за часом,  $U$  – швидкість.

Це означає, що швидкість поширення фізичних збурень  $v$  не повинна перевищувати швидкість поширення численних збурень  $\delta x / \delta t$ . Для стійкості явної різницевої схеми необхідно, щоб область залежності розв'язку диференціальної задачі знаходилася би всередині області залежності розв'язку різницевої задачі. Швидкість потоку змінюється по всьому домену, і ми повинні забезпечити  $Co < 1$  усюди. Тому ми вибираємо крок за часом  $\delta t$  на основі найгіршого випадку: максимальне число Куранта  $Co_{\max} < 1$ .

### 1.1.4 Обробка та візуалізація результатів розрахунку (Postprocessing)

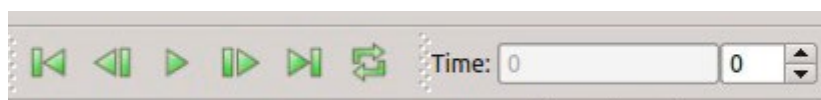
Візуалізація результатів обчислень і виконується у графічному інтерфейсі ParaView, що поставляється разом з пакетом OpenFOAM. Пакет запускається командою **paraFoam** з каталогу розрахункової задачі, автоматично формується пустий файл OpenFOAM. Після натискання **APPLY** в інтерфейсі ParaView завантажується геометрія задачі.

Обираємо поле змінної величини (тиск  $p$  або швидкість  $U$ ):



Для вектора швидкості можна побудувати поля модуля швидкості та окремих компонентів. Surface означає гладке поле (рис. 1.1.3, а), Surface with Edges – поле з нанесеною на нього сіткою (рис. 1.1.3, б).

Еволюцію картини течії можна побачити, використовуючи інструмент:



Змінити кольорову гамму та шкалу за допомогою інструментів:

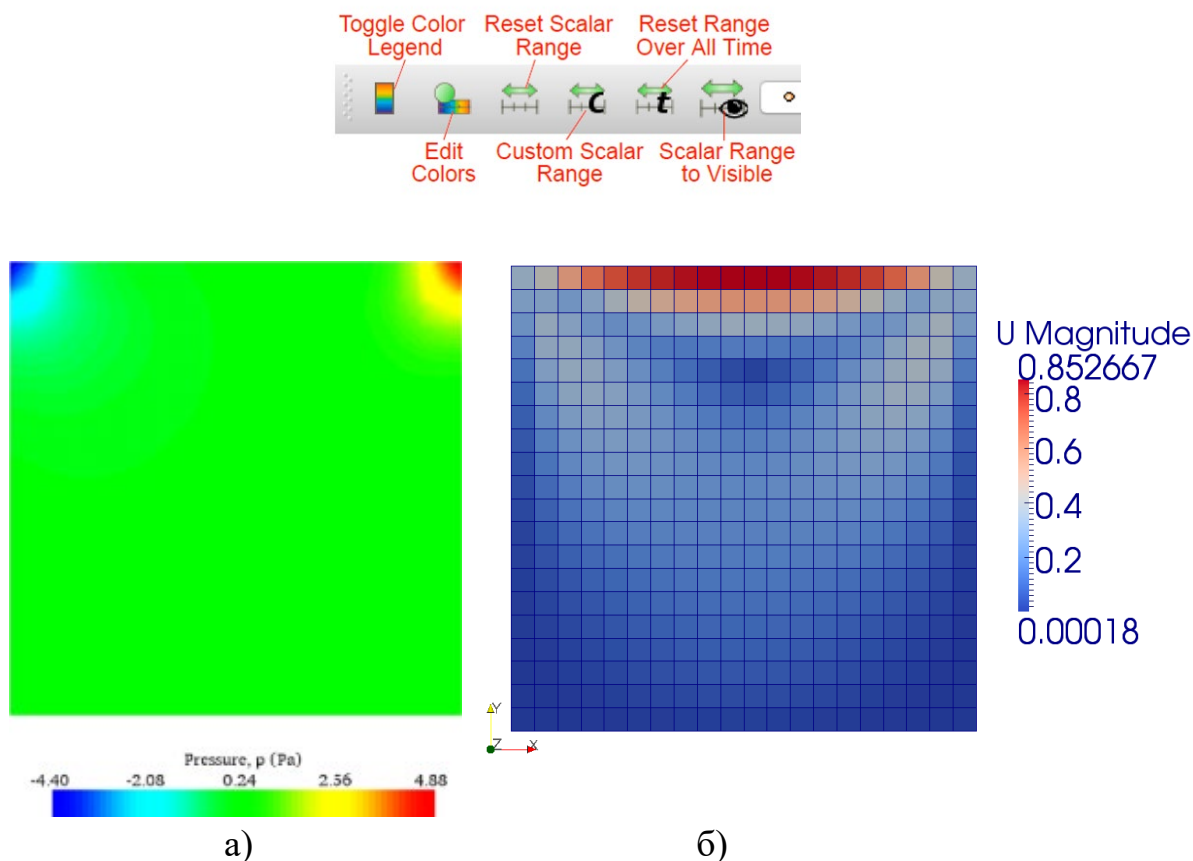
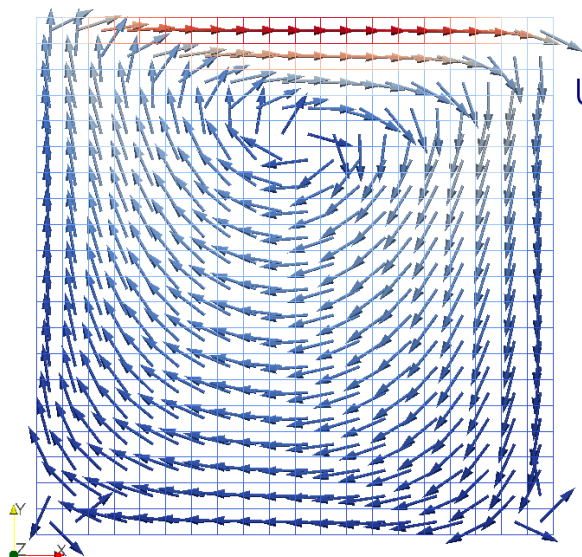


Рис. 1.1.3 Поля змінних фізичних величин: а) тиск; б) модуль швидкості

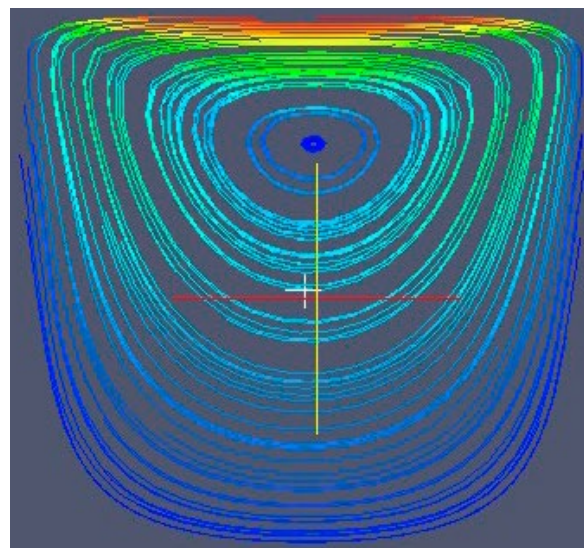
Для побудови векторного поля та ліній течії попередньо витягаємо центри комірок. Цю процедуру рекомендовано проводити у зв'язку з тим, що OpenFOAM обчислює поля в центрах комірок, а Paraview автоматично інтерполює значення на центри вузлів.

- 1) Центри комірок: **Filters** → **Alphabetical** → **Cell Centers**.
- 2) Будуємо вектори: **Filters** → **Alphabetical** → **Glyphs** (рис. 1.1.4).
- 3) Будуємо лінії течії: **Filters** → **Alphabetical** → **Stream Tracer** (рис. 1.1.5).



а)

Рис. 1.1.4 Вектори



б)

Рис. 1.1.5 Лінії течії

### Завдання

1. Завантажити задачу:  
**cd \$FOAM\_RUN/tutorials/incompressible/icoFoam/cavity/cavity**
2. Порядок виконання (команди):
  - Будування сітки **blockMesh;**
  - Перевірка сітки **checkMesh;**
  - Запуск розрахунку **icoFoam;**
  - Візуалізація результатів **paraFoam.**
3. Самостійне завдання:
  - Зміна геометричних параметрів сітки: **system/blockMeshDict;**
  - Зміна числа Рейнольдса за рахунок зміни:
    - швидкості течії: **0/U;**
    - коефіцієнта кінематичної в'язкості: **constant/transportProperties**



## Робота 1.2. Розрахунок стаціонарної течії в каналі з уступом.

Лекційний матеріал. Розділ 3-7

Література: [4-15]

### 1.2.1 Постановка задачі

Задача: \$FOAM\_TUTORIALS/incompressible/simpleFoam/pitzDaily

Посилання: <https://www.openfoam.com/documentation/tutorial-guide/tutorialse5.php>

Досліджується стаціонарна двовимірна ізотермічна турбулентна течія нестисливої однорідної ньютонівської рідини в каналі зі зворотнім уступом. Геометрію задачі представлено на рис. 1.2.1. Вона відповідає експериментальному дослідженню R.W.Pitz та J.W.Daily [15].

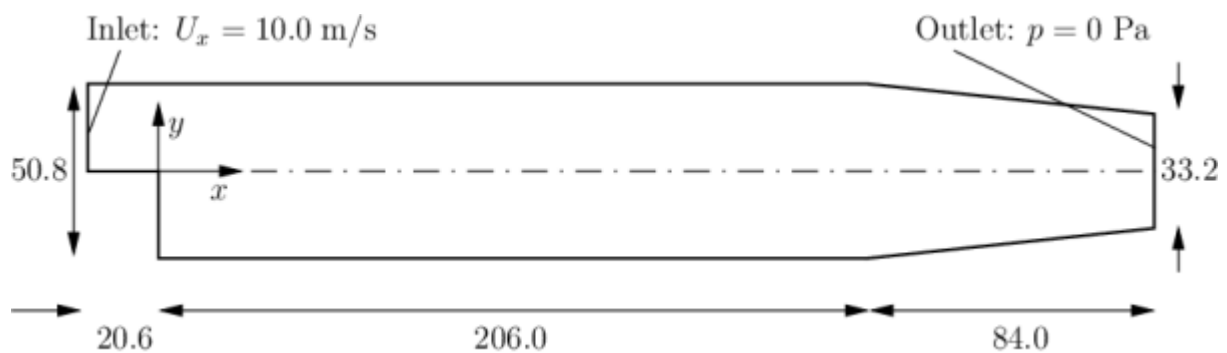


Рис. 1.2.1 Геометрія задачі (розміри в мм)

Система визначальних рівнянь складається з рівняння Нав'є-Стокса нерозривності і записується наступним чином:

$$(\mathbf{U} \cdot \nabla) \mathbf{U} + \nabla \cdot \mathbf{R} = -\nabla p,$$

$$\nabla \cdot \mathbf{U} = 0,$$

де  $\mathbf{U} = \{U_x, U_y, U_z\}$  – вектор швидкості,  $p$  – кінематичний тиск,  $\mathbf{R} = \nu_{\text{eff}} \nabla \mathbf{U}$  – в'язкий тензор напруг,  $\nu_{\text{eff}}$  – ефективний (модельний) коефіцієнт кінематичної в'язкості, що залежить від обраної моделі турбулентності.

## 1.2.2 Підготовка вихідних даних (Preprocessing)

### 1.2.2.1 Розрахункова сітка

Очікується, що потік в досліджуваному каналі буде мати досить складну вихрову структуру, і для оптимального розв'язку будуть потрібні згущення сітки. Області з великими градієнтами фізичних величин вимагають більш дрібної сітки, ніж низькоградієнтні. Такими областями в першу чергу є примежовий шар і зони вихорів. Отже, згущення сітки необхідні близько до центральної осі, особливо за уступом, і близько до стінок каналу.

Для побудови блочно-структурованої сітки використовуємо утіліту **blockMesh**. Структуру блоків показано на рис. 1.2.2.

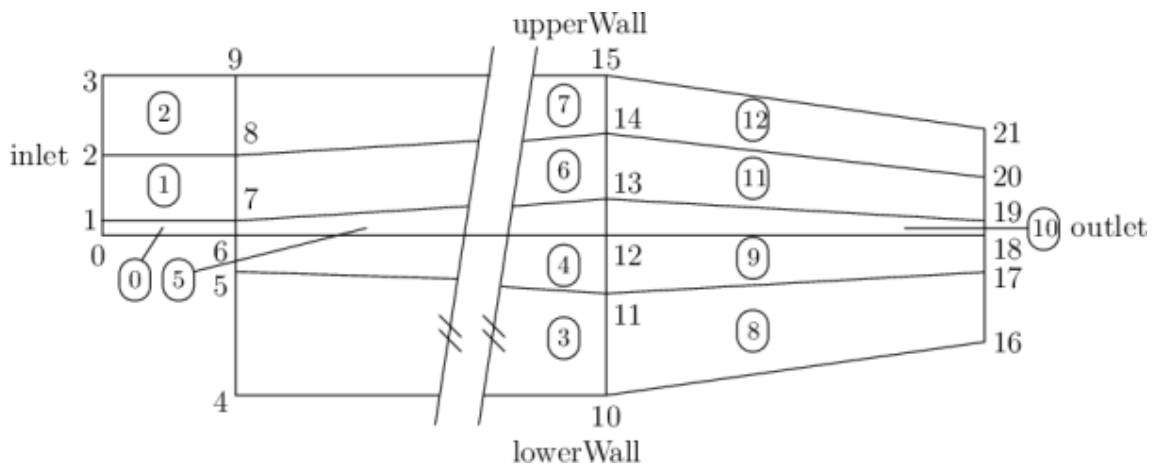


Рис. 1.2.2 Структура блоків

Повний опис структури сітки (координати вершин, поділ на блоки, параметри розбиття, тощо) надано в файлі `system/blockMeshDict`. Після виконання утіліти **blockMesh** рекомендується перевірка сітки на наявність помилок утілітою **checkMesh**.

### 1.2.2.2 Крайові умови

Початкові та граничні умови задаються в каталозі 0 для кожної змінної фізичної величини. Для даної задачі турбулентної течії нестисливої рідини змінними являються тиск  $p$ , вектор швидкості  $\mathbf{U}$ :

- Початкові умови:

- $\mathbf{U} = 0 \text{ м/с}$ ,  $p = 0 \text{ м}^2/\text{с}^2$  – необхідні у вхідних файлах OpenFOAM, але не потрібні для розв'язку, оскільки поставлена задача стаціонарна.

- Граничні умови:
  - inlet – на вході в канал задано постійну швидкість  $U = (10\ 0\ 0)$  м/с і нульовий градієнт тиску;
  - outlet – на виході з каналу – нульовий градієнт швидкості (тобто, вихід всіх збурень) та нульовий тиск  $p = 0$ ;
  - upperWall та lowerWall – умова прилипання на стінках;
  - frontAndBack – пусті граничні умови для 2D задачі, щоб виключити обчислення в третьому вимірі.

Крім того, необхідно задати крайові умови для змінних фізичних величин, що відносяться до обраної моделі турбулентності. Наприклад, для моделі  $k$ - $\epsilon$  додатковими змінними є кінетична енергія турбулентності  $k$ , швидкість дисипації енергії турбулентності  $\epsilon$  та модельний коефіцієнт турбулентної в'язкості  $\nu_t$ . Асимптотичні оцінки граничних умов  $k$ - $\epsilon$  моделі надано в Розділі 2.1.8.1 Посібника користувача OpenFOAM [5].

### 1.2.2.3 Фізичні властивості середовища

Дана задача є нестисливою, тому достатньо вказати тільки коефіцієнт кінематичної в'язкості  $\nu$  в файлі constant/transportProperties.

### 1.2.2.4 Модель турбулентності

Модель турбулентності задається в файлі constant/turbulenceProperties:

```
simulationType RAS; // Клас моделі турбулентності: RAS, LES або laminar

RAS
{
    RASModel      kEpsilon; // назва моделі турбулентності (в даному випадку k-ε)

    turbulence    on;

    printCoeffs   on;
}
```

Повний список моделей турбулентності, які вбудовано в пакет OpenFOAM можна переглянути в Розділі 7.2 Посібника користувача [5]. Також список доступних моделей турбулентності, наприклад, для чисельної моделі simpleFoam можна отримати командою:

**simpleFoam -listTurbulenceModels**

### 1.2.3 Чисельна модель

Використовується чисельна модель (solver) **simpleFoam**, програмні коди якої можна знайти в каталозі <.../applications/solvers/incompressible/simpleFoam>

#### 1.2.3.1 Налаштування чисельних схем

Інформацію про дискретизацію системи визначальних рівнянь можна знайти у файлі `system/fvSchemes`. Головна особливість стаціонарної задачі полягає у відсутності похідної за часом, проте, пакет OpenFOAM вимагає вказати схему дискретизації для неї:

```
ddtSchemes // перша похідна за часом
{
    default    steadyState; // для всіх фізичних величин задано стаціонарний стан
}
```

#### 1.2.3.2 Налаштування методів розв'язку СЛАР

У файлі `system/fvSolutions` описано методи розв'язку системи лінійних алгебраїчних рівнянь (СЛАР), значення абсолютної і відносної похибки.

```
solvers
{
    p
    {
        solver      GAMG; // геометро-алгебраїчний багатосітковий метод розв'язку СЛАР
        tolerance    1e-06; // абсолютна нев'язка
        relTol       0.1; // відносна нев'язка
        smoother     GaussSeidel; // метод згладжування Гаусса-Зейделя
    }

    "(U|k|epsilon|omega|f|v2)"
    {
        solver      smoothSolver;
        smoother     symGaussSeidel;
        tolerance    1e-05;
        relTol       0.1;
    }
}
```

Для поля тиску використовується геометро-алгебраїчний багатосітковий метод (GAMG), який можна застосовувати незалежно від симетричності

матриці системи. Для всіх інших полів фізичних величин (швидкість  $U$  та змінні обраної моделі турбулентності, зокрема енергія турбулентності  $k$ , швидкість дисипації  $\varepsilon$  та ін.) використовується ітераційний алгоритм smoothSolver, що використовує метод Гаусса-Зейделя для симетричних матриць, щоб розв'язок зійшовся до необхідної точності (tolerance).

Особливої уваги потребує процедура сумісної корекції полів швидкості та тиску SIMPLE (Semi-Implicit Method for Pressure Linked Equations) – напівнеявний метод, який використовується для стаціонарних задач, виконує тільки 1 корекцію. Якщо стаціонарна задача вирішується ітеративно, не потрібно повністю розв'язувати лінійну залежність швидкості та тиску, оскільки зміни між послідовними розв'язками вже не малі. Нелінійність системи стає важливішою, оскільки ефективний крок часу значно більший. Алгоритм SIMPLE сформульований, спираючись на такі гіпотези:

- Апроксимація поля швидкості отримується шляхом розв'язання рівняння імпульсу. Член градієнта тиску розраховується, використовуючи розподіл тиску за попередньою ітерацією або початкові умови.
- Рівняння тиску формулюється та розв'язується для отримання нового розподілу тиску.

### 1.2.3.3 Розрахунковий експеримент

Контроль за ходом розрахунків здійснюється в файлі system/controlDict. У цьому файлі прописуються:

```
application    simpleFoam; // чисельна модель (solver)
startFrom      startTime; // початковий момент часу
startTime      0;
stopAt         endTime;   // кінцевий момент часу
endTime        2000;
deltaT         1;         // крок за часом
writeControl    timeStep;
writeInterval   100;      // крок виведення результатів
writePrecision  6;        // точність виведення результатів
```

**Зверніть увагу!** В стаціонарних задачах мова йдеться про «фіктивний» час, тобто номер ітерації. Розрахунок закінчиться автоматично, коли отриманий розв'язок зійдеться з заданою точністю.

### 1.2.4 Обробка та візуалізація результатів розрахунку (Postprocessing)

Візуалізація результатів обчислень і виконується у графічному пакеті ParaView за тим самим алгоритмом, як і в попередній роботі комп'ютерного практикуму 1.1.

В результаті обчислень отримаємо стаціонарну вихрову течію в плоскому каналі зі зворотнім уступом, що представлена на рис. 1.2.3.

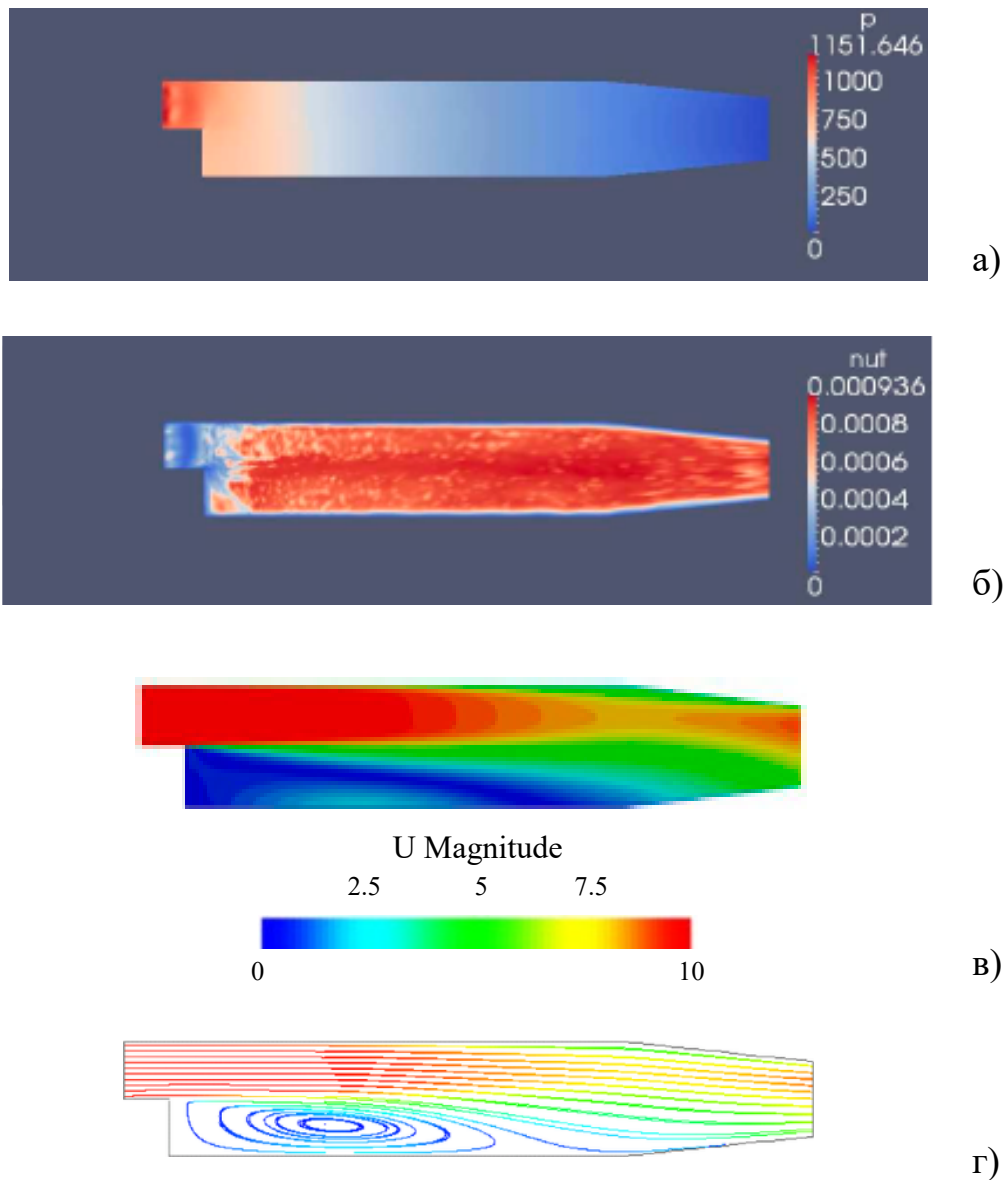


Рис. 1.2.3 Поля змінних фізичних величин:  
а) тиск; б) коефіцієнт турбулентної в'язкості;  
в) модуль швидкості; г) лінії течії

## Завдання

1. Завантажити задачу:  
**cd \$FOAM\_TUTORIALS/incompressible/simpleFoam/pitzDaily**
2. Порядок виконання (команди):
  - Будування сітки **blockMesh;**
  - Перевірка сітки **checkMesh;**
  - Запуск розрахунку **simpleFoam;**
  - Візуалізація результатів **paraFoam.**
3. Самостійне завдання:
  - Зміна геометричних параметрів сітки: **system/blockMeshDict;**
  - Зміна числа Рейнольдса за рахунок зміни:
    - швидкості течії: **0/U;**
    - коефіцієнта кінематичної в'язкості: **constant/transportProperties**

### **Робота 1.3. Будування неструктурованої розрахункової сітки.**

Лекційний матеріал. Розділ 4

Література: [1-3, 5, 16]

**Розрахункова (обчислювальна) сітка** – це дискретне представлення геометричної області, в якій розв’язується задача. Іншими словами, сукупність точок (сіткових вузлів), що задані в області визначення деякої функції.

Розрахункові сітки використовуються при чисельному розв'язанні диференціальних і інтегральних рівнянь. Якість побудови розрахункової сітки в значній мірі визначає успіх або невдачу чисельного розв'язку рівняння.

Процедуру побудови розрахункової сітки можна розглядати як побудову взаємно-однозначного відображення області визначення функції (фізичної області) на деяку розрахункову область, що має більш просту форму.

**Неструктурована (нерегулярна) сітка** – частина евклідової площини або евклідового простору, що розбита на прості фігури, такі як трикутники або тетраедри, в нерівномірній формі. Набір вузлів нерегулярної сітки зовні виглядає хаотично.

*Переваги:*

- підходить для областей довільної геометрії;
- немає обмежень на форму і кількість сусідніх елементів;
- можливість локального подібнення.

*Недоліки:*

- нерегулярність структури даних, відповідно більш складні і повільні алгоритми розв'язку.

#### **1.3.1 Постановка задачі**

В даній роботі комп'ютерного практикуму пропонується побудувати неструктуровану розрахункову сітку навколо тривимірного тіла або декількох тіл, що обтікаються суцільним середовищем, в пакеті SALOME [1]. Як приклад розглянемо задачу обтікання двох тіл: тор і сфера (рис. 1.3.1).



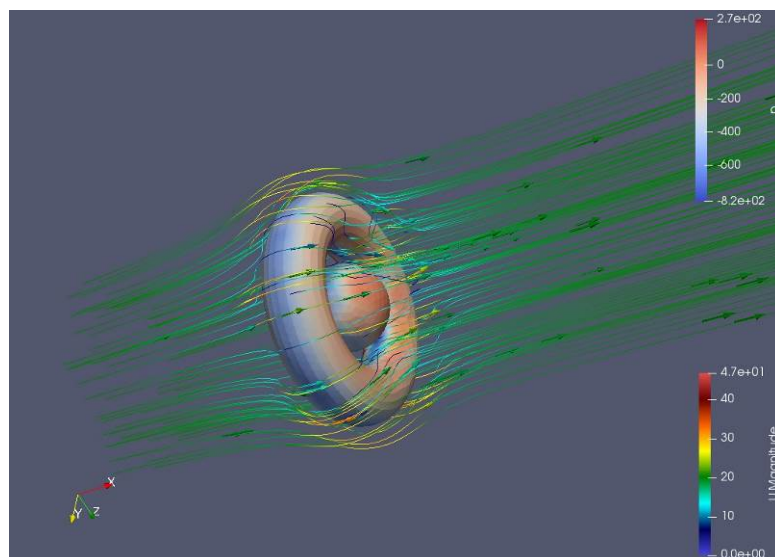


Рис. 1.3.1 Результати розрахунків задачі обтікання двох тіл: тора і сфери

### 1.3.2 Побудова геометрії в пакеті SALOME (модуль Geometry)

Мета модуля **Geometry** – підготовка геометрії розрахункової області. Структуру інтерфейсу можна бачити на рис. 1.3.2. [16]

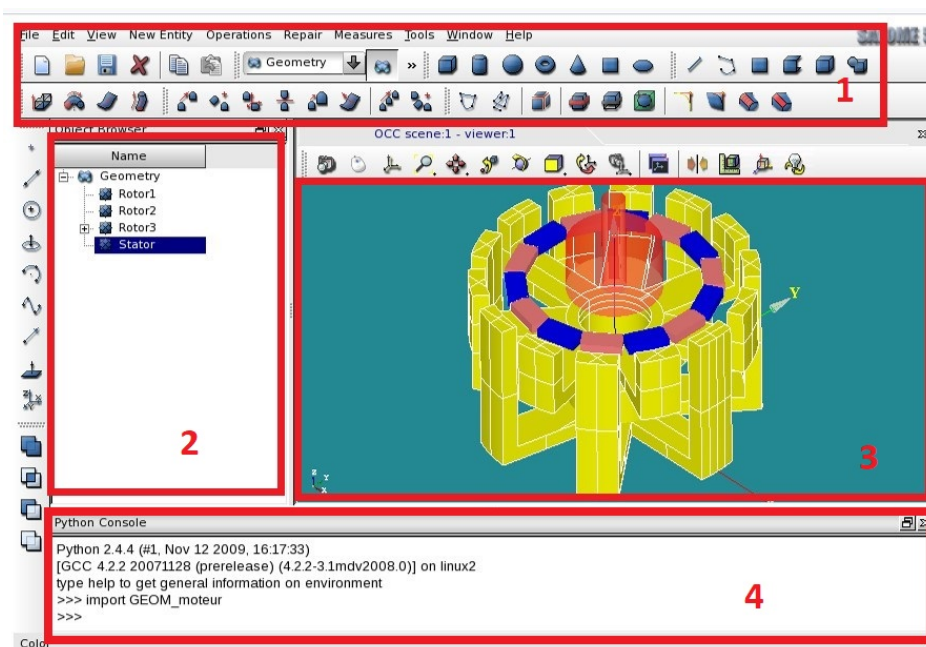


Рис. 1.3.2 Структуру інтерфейсу модуля Geometry: меню і панель інструментів (1), дерево об'єктів (2), основне вікно (3), вікно повідомлень системи (4)

Не можна розглядати цей модуль як виключно креслярський інструмент. Вимоги до геометрії визначаються вимогами до розрахункової сітки (структурована/неструктурована, тип елементів). Геометрія може бути створена як повністю в SALOME, так і імпортована з інших програм (SolidWorks, AutoCAD Mechanical Desktop, ICEM CFD), формати: IGES, STEP, BREP. Можливо також імпортувати геометрію з будь-якого іншого формату з використанням інтерфейсу до мови Python.

### 1.3.2.1 Параметризація геометрії (Notebook)

SALOME Notebook дозволяє визначати змінні параметри, які будуть використовуватися для створення та модифікації геометричних об'єктів (рис. 1.3.3). Інструмент Notebook не обов'язковий для використання, але значно спрощує перебудування геометрії розрахункової області при зміні розмірів і взаємному розташуванню об'єктів. Також підтримуються прості математичні операції (додавання, віднімання, множення і т.п.).

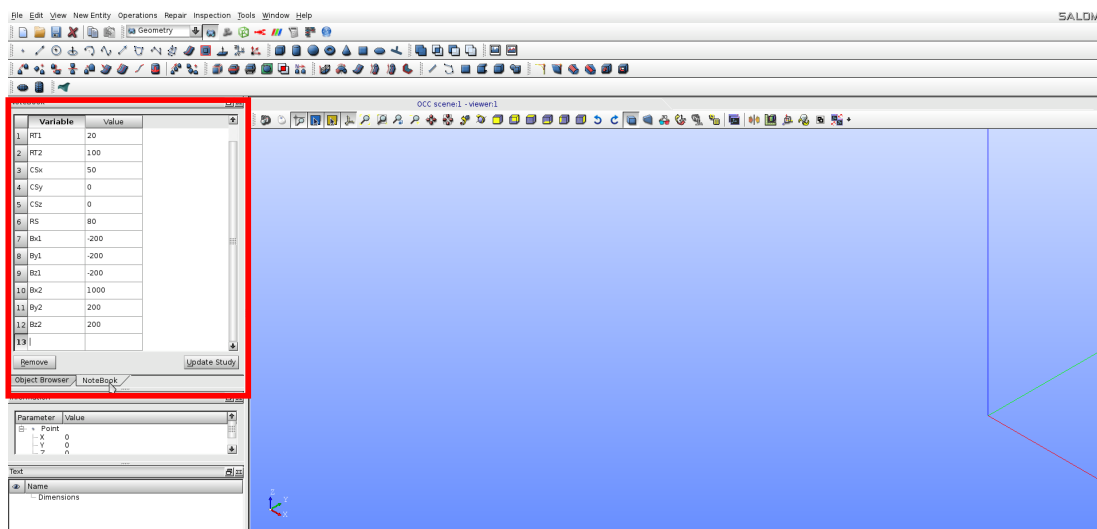


Рис. 1.3.3 Параметризація геометрії (Notebook)

### 1.3.2.2 Геометрія розрахункової області

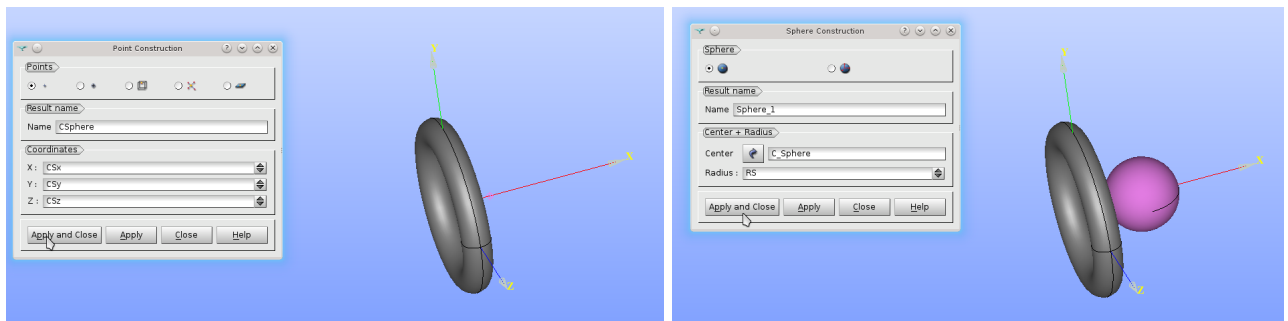
Для побудови геометрії розрахункової області будемо використовувати метод «зверху вниз», що передбачає створення тривимірних геометричних об'єктів з наявних шаблонів та логічні операції між ними.

Список примітивних 3D шаблонів можна знайти в меню:

### New Entity → Primitives



Таким чином, згідно до постановки задачі створюємо два об'єкти, що будуть обтікатися суцільним середовищем: тор і сфера (рис. 1.3.4).



а)

б)

Рис. 1.3.4 Об'єкти, що обтікаються: тор (а) і сфера (б)

Центр сфери знаходиться не на початку координат, тому попередньо необхідно побудувати точку C\_Sphere, що вказана в якості центру на рис. 1.3.4 а). Список базових примітивів можна знайти в меню:

### New Entity → Basic

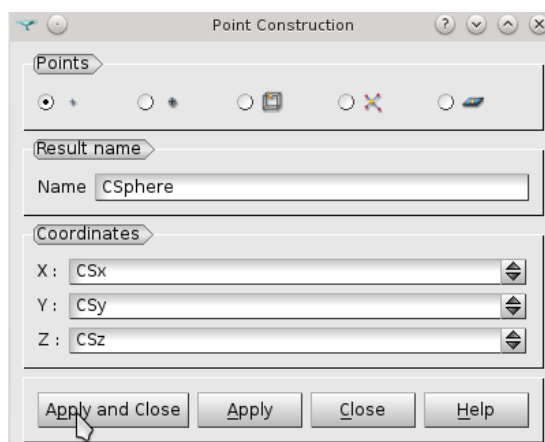


Рис. 1.3.5 Центр сфери

Пакет SALOME дозволяє будування точки декількома способами (рис. 1.3.5):

- по заданих координатах;
- по іншій точці і заданому зміщенню;
- по розташуванню точки на ребрі (середина, край, тощо);
- на перетині ребер;
- на поверхні з параметричним завданням положення точки на цій поверхні.

Ми скористаємося найпростішим способом – задамо координати з таблиці параметрів на рис. 1.3.3. Можна записати координати точки цифрами без використання параметризації. Але в такому разі змінити геометрію розрахункової області змінити вже неможливо.

Для зручності, особливо при роботі зі складною геометрією, можна об'єднати два тіла в одну групу Compound, як показано на рис.1.3.6. Це дозволяє спростити подальшу роботу з ними, наприклад, коли будемо виділяти обтічні поверхні під єдиною назвою та типом.

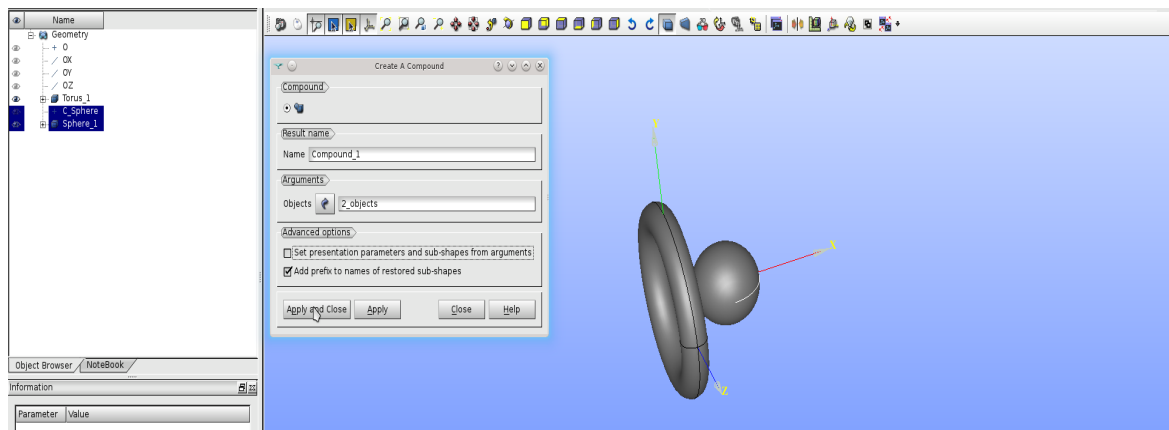


Рис. 1.3.6 Об'єднання сфери та тора в групу

Для побудови паралелепіпеда, що зображено на рис. 1.3.7, достатньо попередньо задати дві точки з координатами (Bx1,Bu1,Bz1) та (Bx2,Bu2,Bz2) згідно до таблиці параметрів на рис. 1.3.3. Паралелепіпед є в списку примітивних шаблонів:

**New Entity → Primitives → Box**

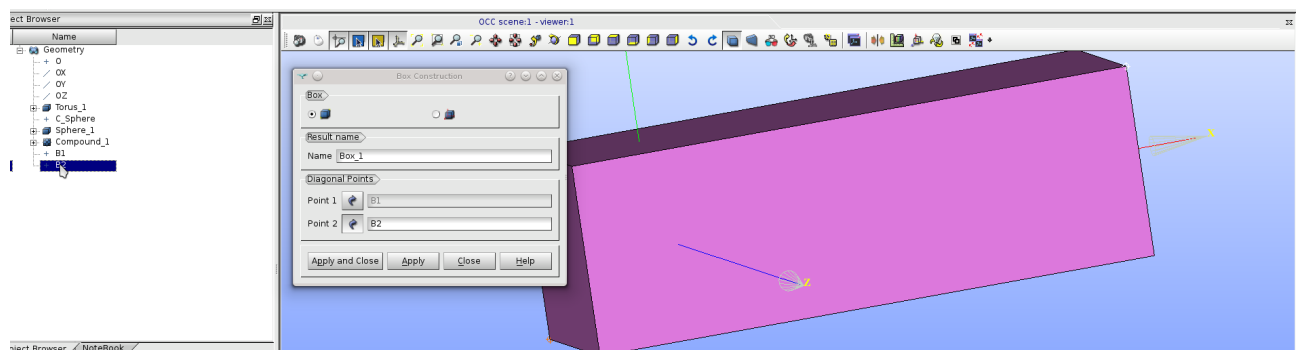


Рис. 1.3.7 Паралелепіпед, що побудовано через дві точки

Розрахункова область поставленої задачі представляє собою паралелепіпед з вирізаними з нього обтічними тілами, як показано на рис. 1.3.8.

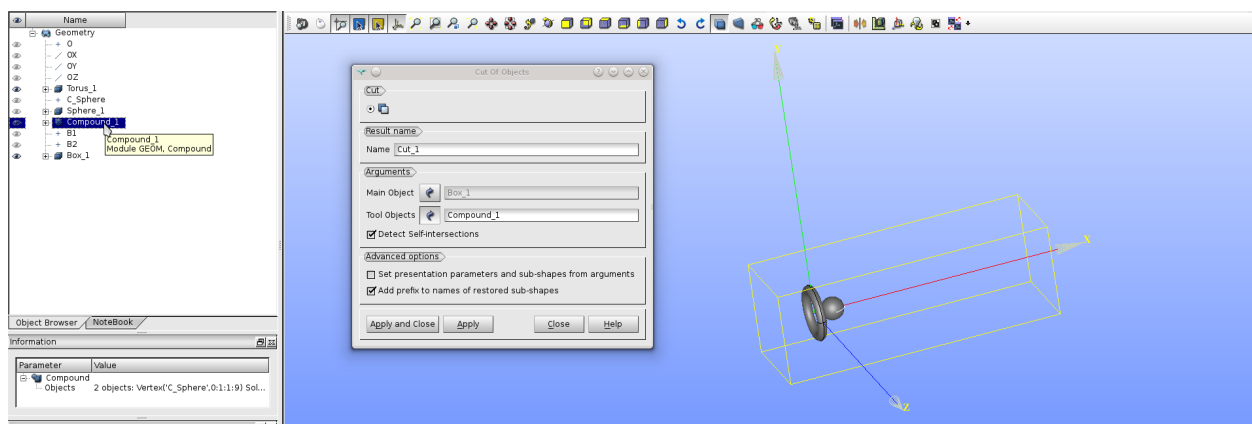


Рис. 1.3.8 Розрахункова область

Операцію «вирізати» можна знайти в меню на панелі управління:

**Operations → Boolean → Cut**



Пакет SALOME містить наступний набір логічних операцій:

- Fuse – склеює об'єкти;
- Common – логічне «і», результатом цієї операції буде область, що містить обидва об'єкти, все інше вирізається;
- Cut – вирізає один об'єкт або групу фігур із іншого;
- Intersection – виконує перетин між двома фігурами.

### 1.3.2.3 Граничні поверхні

Останнім і дуже важливим етапом побудови геометрії розрахункової області є виділення граничних поверхонь. По-перше, на них будуть задаватися граничні умови в пакеті OpenFOAM. Тому, назви поверхонь в пакеті SALOME мають повністю відповідати назвам поверхонь в задачі OpenFOAM в папці 0.

Граничні поверхні (рис. 1.3.9) задаються шляхом виділення частини тривимірної розрахункової області – двовимірної поверхні.

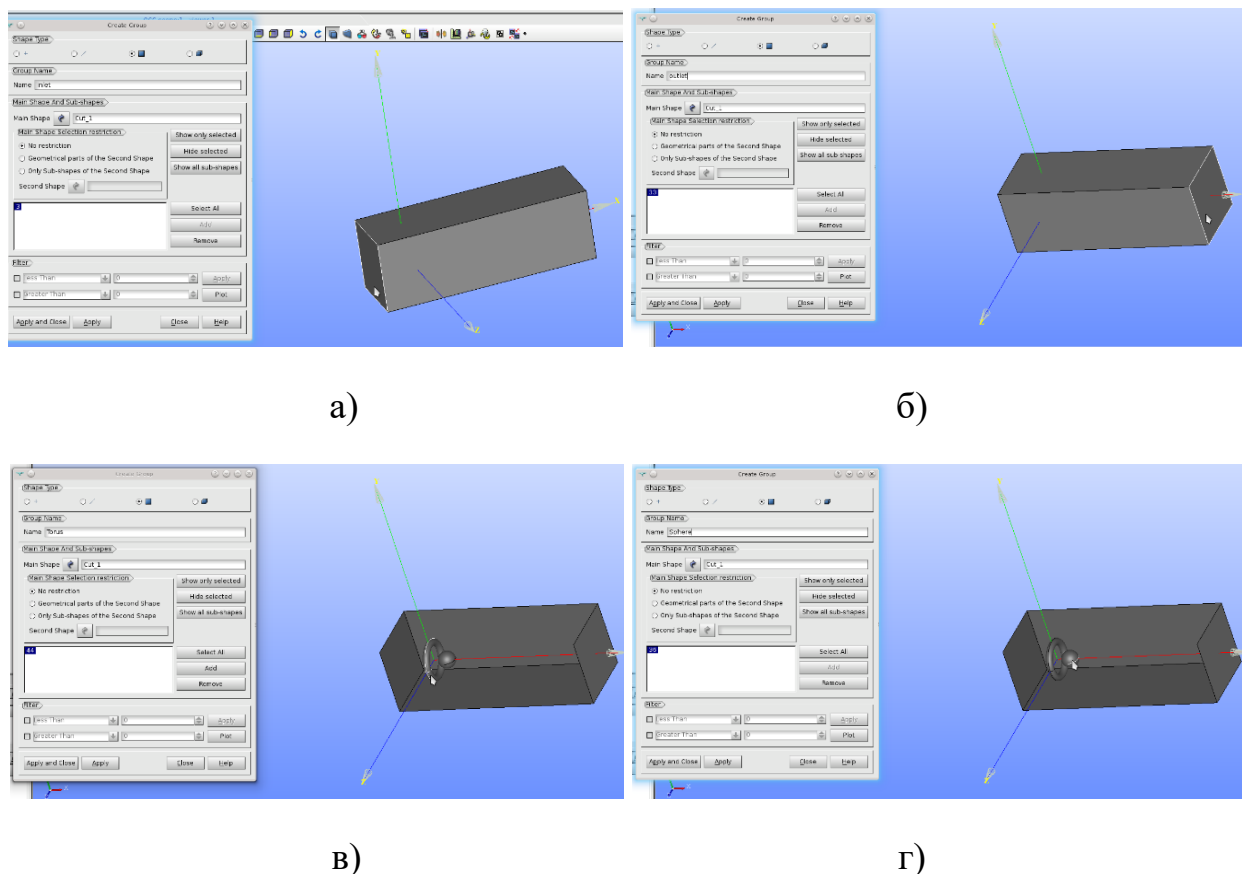


Рис. 1.3.9 Граничні поверхні: inlet (а), outlet (б), Torus (в), Sphere (г)

Таким чином, необхідно створити групу об'єктів в меню:

### **New Entity → Group → Create Group**

Тип об'єкту (Shape Type) обрати 2D поверхню. Виділяються фігури простим натисканням і додаванням (кнопка Add). Приховані граничні поверхні, як наприклад, Torus на рис. 1.3.9, в), можна виділити, приховавши інші кнопкою Hide selected.

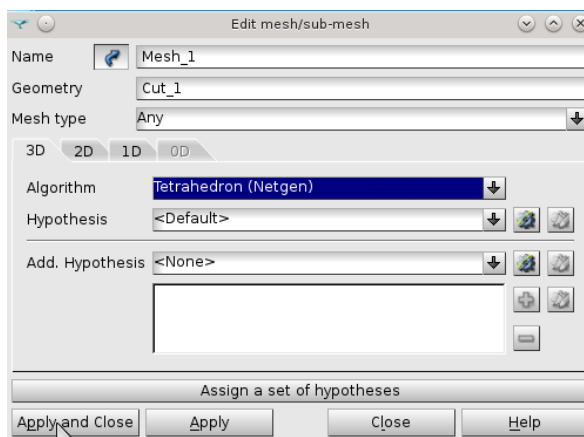
Треба зауважити, що поверхні Torus (рис. 1.3.9, в) та Sphere (рис. 1.3.9, г) доцільно було б об'єднати в одну, наприклад, wall, тому що на них планується задавати однакові граничні умови прилипання.

## **1.3.3 Побудова розрахункової сітки в пакеті SALOME (модуль Mesh)**

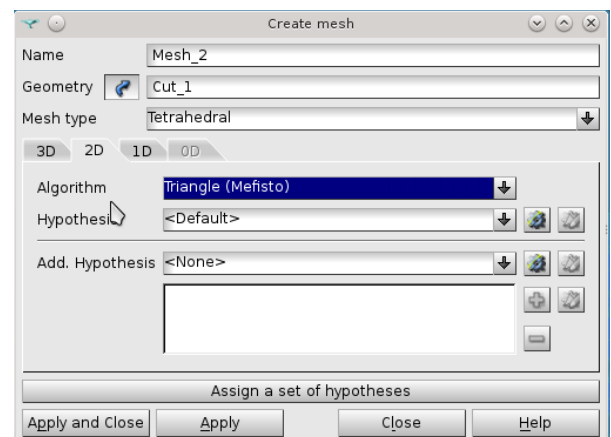
### **1.3.3.1 Основна сітка**

Створення сітки запускається командою Create Mesh в розділі Mesh панелі управління. Відкривається менеджер створення базової сітки. У ньому присутні такі поля:

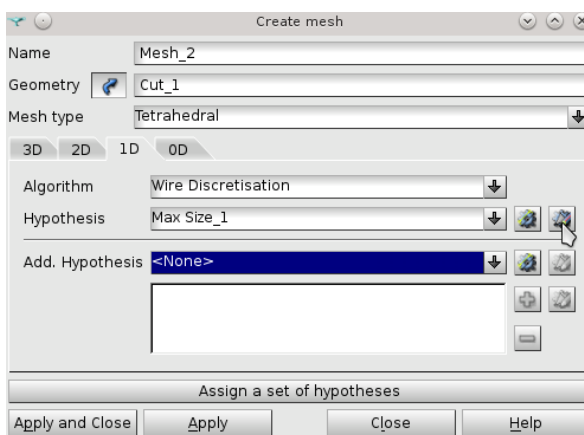
- Name - ім'я сітки.
- Geometry – об'єкт з розділу Geom, на основі якого будується сітка. Об'єкт Cut в нашому випадку.
- Підрозділи 3D, 2D, 1D і 0D відкривають меню налаштування дискретизації об'єктів різних розмірностей: об'ємів, поверхонь, ребер і вузлів (рис. 1.3.10).
- Кнопка Assign a set of hypotheses – дозволяє швидко налаштувати всі параметри за певним шаблоном.



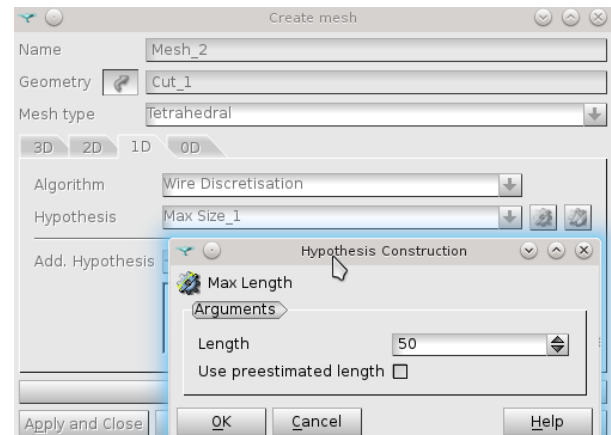
а)



б)



в)



г)

Рис. 1.3.10 Налаштування дискретизації: об'ємів (а), поверхонь (б), ребер (в), гіпотеза дискретизації – максимальний розмір комірки (г)

Командою Compute розраховується сітка з заданими параметрами, як показано на рис.1.3.11. На монітор виводиться статистика: форма і кількість елементів сітки.

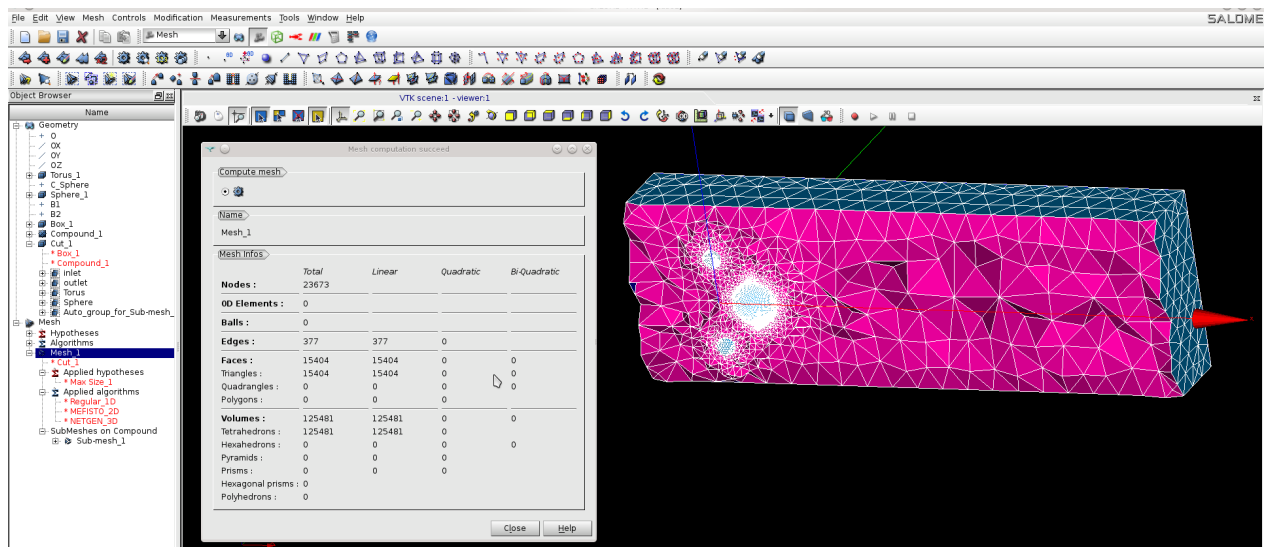


Рис. 1.3.11 Розрахункова сітка

В розділі Control панелі управління можна дослідити більш детальну інформацію про сітку, зокрема:

- наявність вільних границь;
- розподіл довжин ребер;
- вільні вузли;
- вільні ребра;
- площі граней;
- об'єми комірок;
- масштабність
- та ін.

### 1.3.3.2 Підсітки

Під-сітка (Sub-mesh) – це область, яка містить посилання на частину об'єкта (одну або кілька груп всередині геометричного об'єкта або будь-якого його під-елемента) і визначає нові правила розбиття на обраному просторі. Іншими словами, при дискретизації розрахункової області, на тій множині, де визначена під-сітка використовуються алгоритми, гіпотези і додаткові гіпотези, відмінні від тих, що задаються для всієї сітки (Mesh).



В поставленій задачі доречно згустити розрахункову сітку поблизу обтічних тіл, де спостерігаються найбільші градієнти фізичних величин у зв'язку з опором тертя на твердих поверхнях.

Створення підсітки запускається командою Create Sub-mesh в розділі Mesh панелі управління. Меню практично не відрізняється від Create Mesh. Необхідно обрати із дерева об'єктів групи поверхонь Torus та Sphere, на основі яких будується підсітка (рис. 1.3.12).

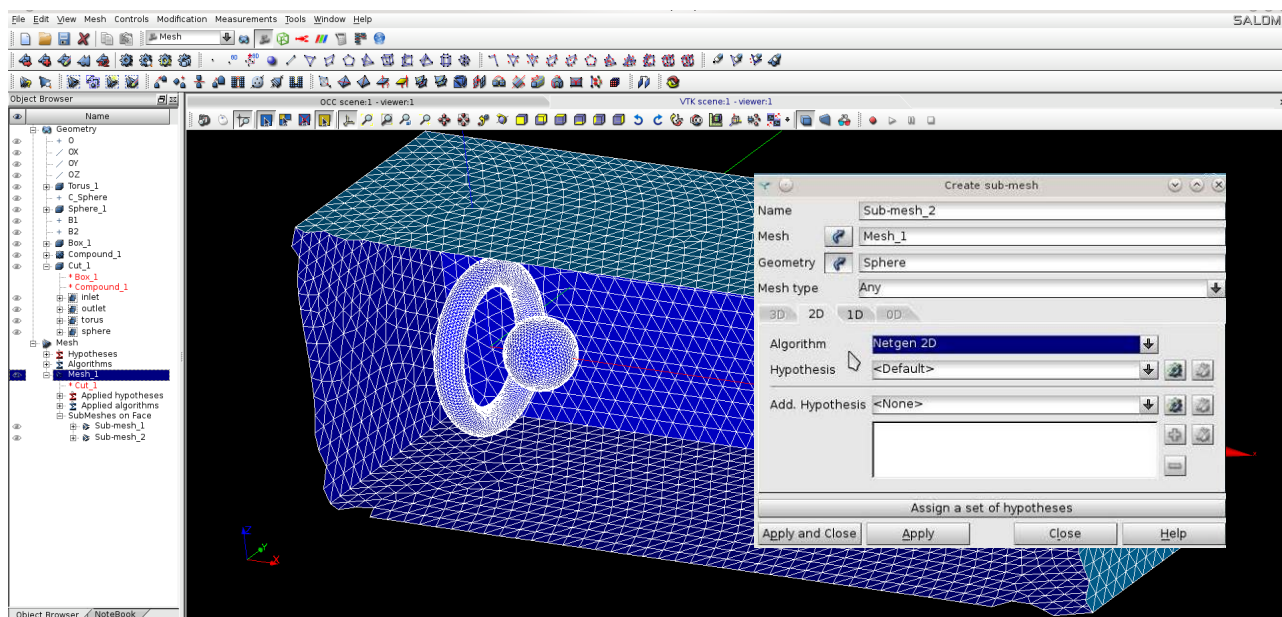


Рис. 1.3.12 Підсітка

Таким чином, отримали тривимірну неструктуровану розрахункову сітку, яку можна бачити на рис. 1.3.11 в перерізі. «Розрізати» сітку можна командою Clipping, яку можна обрати натиснувши на поле правою кнопкою і обрати параметри розрізу.

### 1.3.3.3 Граничні поверхні

Граничні поверхні, що були виділені в модулі Geometry (рис. 1.3.9) просто переносяться на побудовану сітку командою Create Groups From Geometry в меню Mesh. В поле Geometry додаються об'єкти, на основі яких будуються групи елементів (рис. 1.3.13).

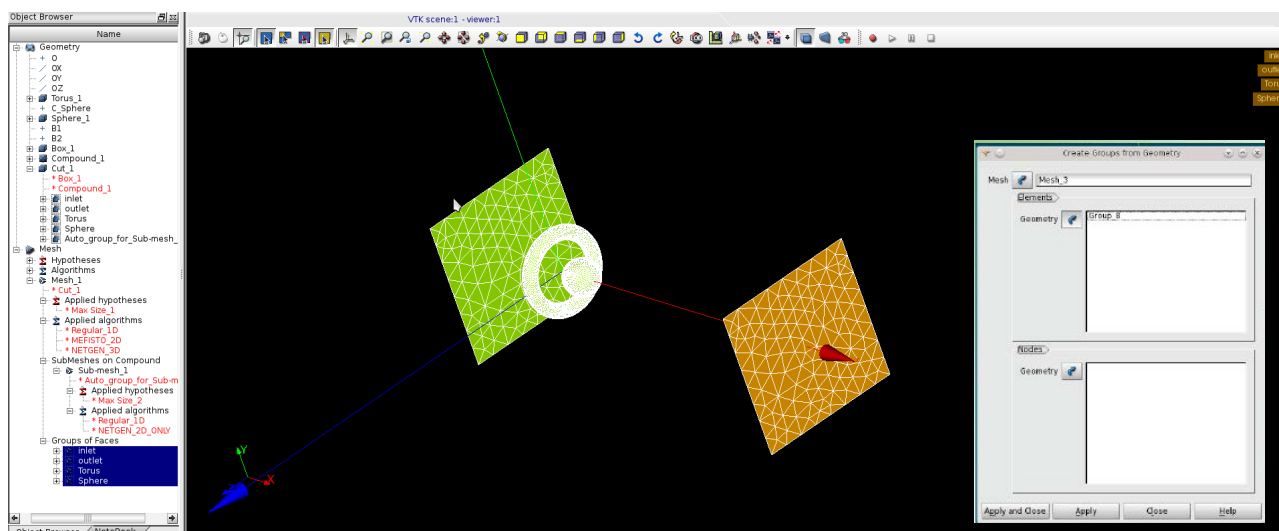


Рис. 1.3.13 Підсітка

Після імпортування розрахункової сітки з пакету SALOME в OpenFOAM невиділені і неназвані поверхні автоматично об'єднуються в одну групу і отримують назву defaultFaces.

**Увага!** Не забувайте зберігати етапи будови геометрії та сітки. Пакет SALOME працює з файлами формату hdf, які можна редагувати. Готову розрахункову сітку необхідно експортувати в формат unv, який вже «розуміє» пакет OpenFOAM.

### 1.3.3 Підготовка задачі в пакеті OpenFOAM

Створена за допомогою Salome сітка і збережена у форматі UNV, конвертується в формат OpenFOAM командою `ideasUnvToFoam mesh.unv`, де `mesh.unv` - ім'я файлу.

Пропонується за основу взяти задачу:

```
cd $FOAM_RUN/tutorials/incompressible/pimpleFoam/RAS/pitzDaily
```

**Увага!** При конвертації сітки граничним поверхням автоматично надається тип `patch` (Таблиця 1). Необхідно самостійно змінити тип поверхні для обтічної сфери і тора на `wall`.

## Завдання

1. Геометрія розрахункової області.
2. Розрахункова сітка.
3. Розрахунок:
  - Експорт сітки в OpenFOAM **ideasUnvToFoam lab3.unv;**
  - Перевірка сітки **checkMesh;**
  - Зміна одиниць виміру (якщо необхідна)  
**transformPoints-scale '(1e-3 1e-3 1e-3)'**
  - Формування файлів задачі (граничні умови, параметри розрахунку);
  - Запуск розрахунку **pimpleFoam;**
  - Візуалізація результатів **paraFoam.**
4. Самостійне завдання:
  - Побудувати розрахункову сітку методом «зверху-вниз», тобто з використанням логічних операцій: 2-3 об'єкта, логічні операції за вибором;
  - Зробити розрахунки в OpenFOAM і візуалізацію результатів в Paraview.

## Робота 1.4. Будування блочно-структурованої розрахункової сітки.

Лекційний матеріал. Розділ 4

Література: [1-3, 5, 16]

**Структуровані (регулярні) сітки** складаються з груп ліній, таких що члени однієї групи не перетинаються між собою і перетинають будь-яку лінію з іншої групи тільки один раз. Структурована сітка логічно еквівалентна декартовій, положення будь-якої точки (або контрольного об'єму) в області унікально визначається набором трьох індексів, наприклад,  $(i, j, k)$

*Переваги:*

- простота опису: один з індексів кожної сусідньої точки  $P$  відрізняється на 1 від відповідного індексу точки  $P$ ;
- під час дискретизації рівнянь в часткових похідних результуюча матриця системи алгебраїчних рівнянь має регулярну структуру, що може бути застосоване під час розробки методу розв'язку.

*Недоліки:*

- застосування тільки для простої геометрії розрахункової області;
- подрібнення сітки в одній області тягне за собою занадто дрібну сітку в інших областях розв'язку і марну трату ресурсів. Довгі вузькі комірки можуть погано впливати на збіжність.

Зважаючи на обмеженість діапазону геометричних об'єктів, що описуються структурованими сітками, і неможливості побудувати єдину сітку для всієї розрахункової області, проводиться поділ на блоки. Складність реалізації такого підходу полягає в процедурі зшивання рішень, отриманих у різних підшарах. Тим не менш, метод побудови блочно-структурованої розрахункової сітки надає широкі можливості для використання ефективних чисельних методів всередині блоків, в кожному з яких генерується окрема сітка регулярної структури.

### 1.4.1 Постановка задачі

В даній роботі комп'ютерного практикуму пропонується побудувати блочно-структуровану розрахункову сітку в пакеті SALOME для класичної задачі обтікання циліндра в двовимірній постановці. На рис. 1.4.1 можна бачити схему розбиття на блоки.

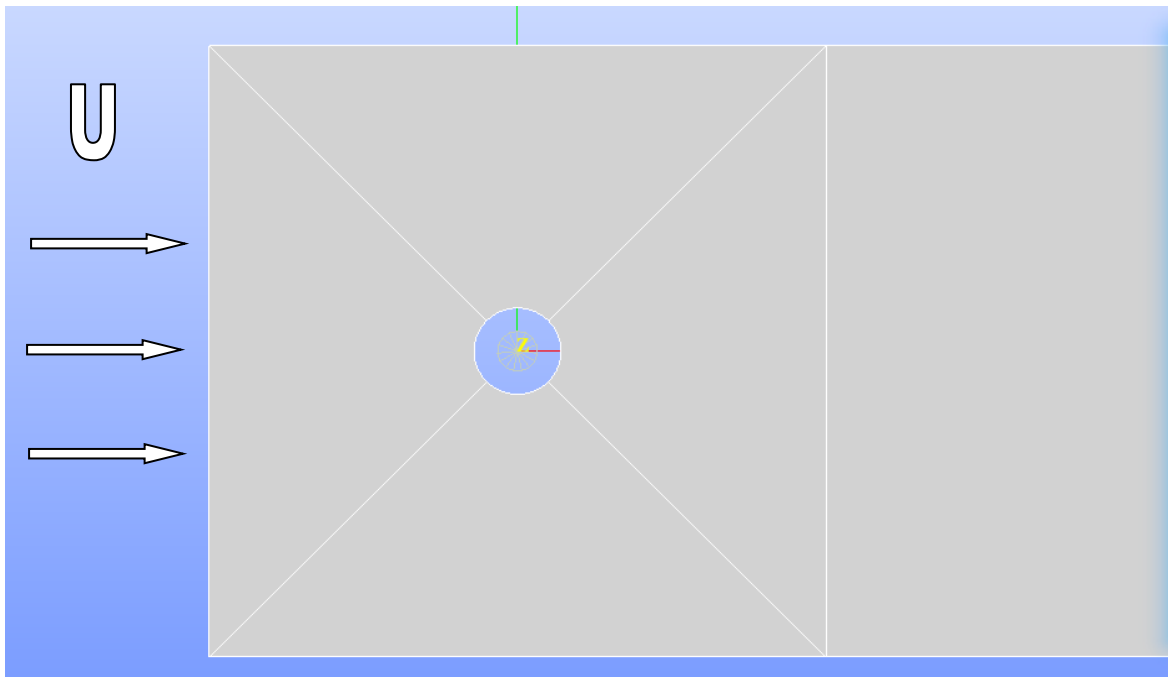


Рис. 1.4.1 Блочна структура геометрії розрахункової області

## 1.4.2 Побудова геометрії в пакеті SALOME (модуль Geometry)

### 1.4.2.1 Параметризація геометрії (Notebook)

NoteBook		
	Variable	Value
1	R1	0.1
2	R2	$10 \cdot R1$
3	L	$5 \cdot R1$
4	z	0.01

Remove Update Study

Object Browser Notebook

Рис. 1.4.2 Параметризація геометрії (Notebook)

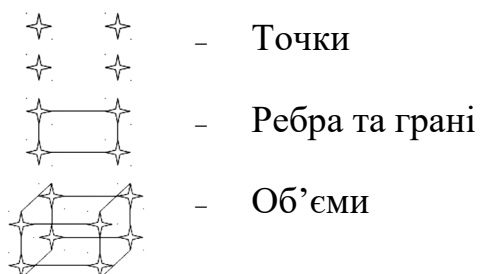
На рис. 1.4.2 представлено змінні параметри, які будуть використані для створення та модифікації геометричних об'єктів.

Інструмент Notebook підтримує прості математичні операції (додавання, віднімання, множення і т.п.). Таким чином, змінивши тільки одну величину R1 (діаметр циліндра), перерахуються інші параметри, що визначають зовнішні розміри розрахункової області.

Параметр z означає лінійний розмір всіх комірок в третьому напрямі, в якому буде виключено розрахунок і задано пусті граничні умови.

### 1.4.2.2 Геометрія розрахункової області

Для побудови геометрії розрахункової області будемо використовувати метод «знизу вверху», що передбачає створення геометричних об'єктів з поетапним збільшенням розмірності:



Таким чином, згідно до постановки задачі необхідно побудувати п'ять блоків, як показано на рис. 1.4.1. Кожен блок відповідає вимогам структурованої гексаедральної сітки: 6 поверхонь, 8 вузлів.

На першому етапі будується коло радіусом  $R_1$  з центром на початку координат (рис. 1.4.3), яке можна знайти в списку базових примітивів:

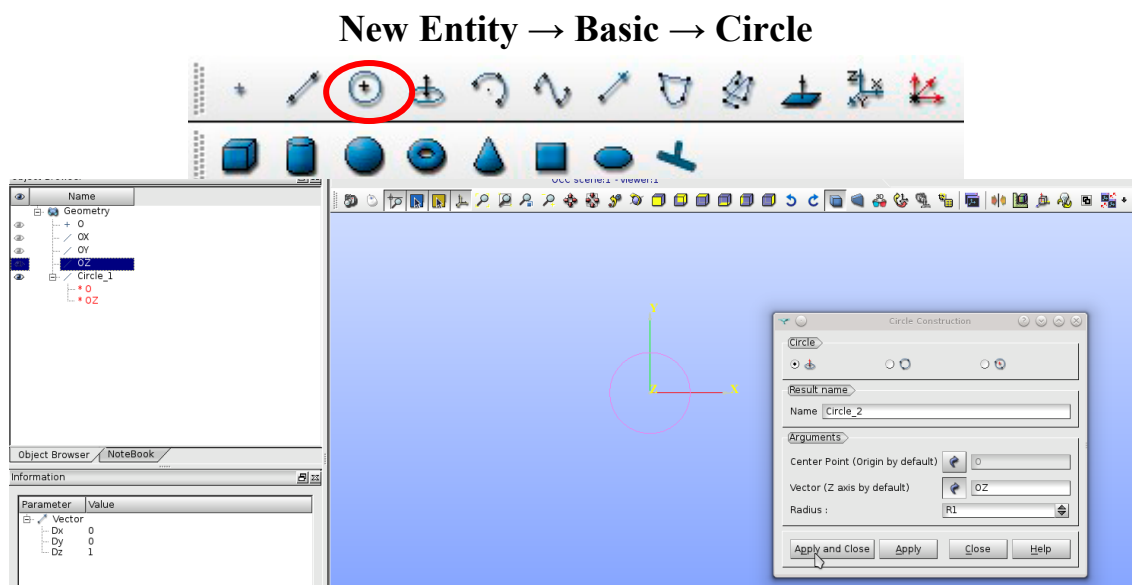
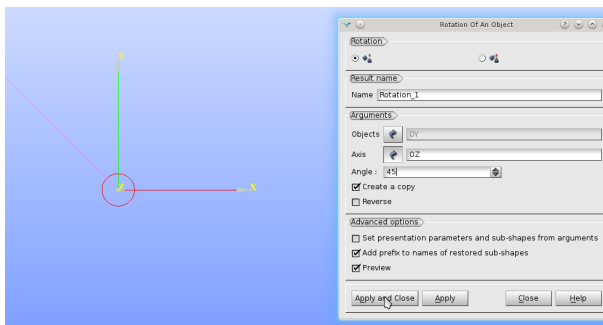


Рис. 1.4.3 Коло з центром на початку координат

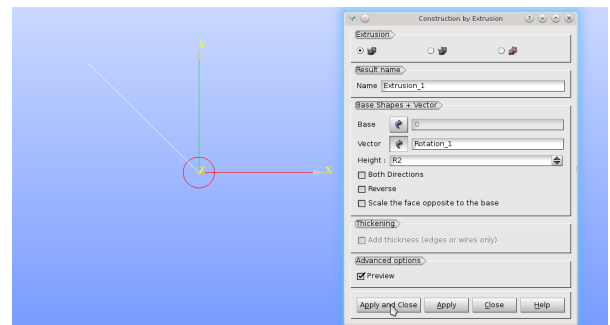
Далі необхідно створити додатковий вектор за допомогою повороту вісі  $OY$  навколо  $OZ$  на  $45^\circ$  (рис. 1.4.4, а).

Список операцій перетворення координат можна знайти в меню:

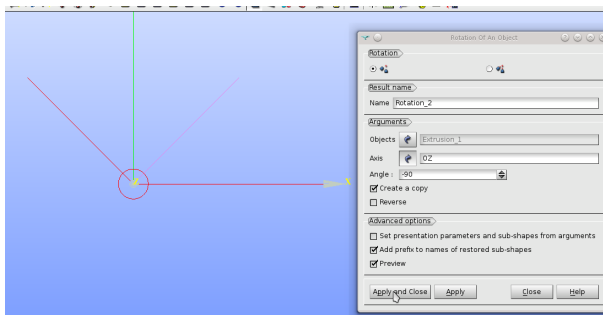
**Operations** → **Transformation** → **Rotation**



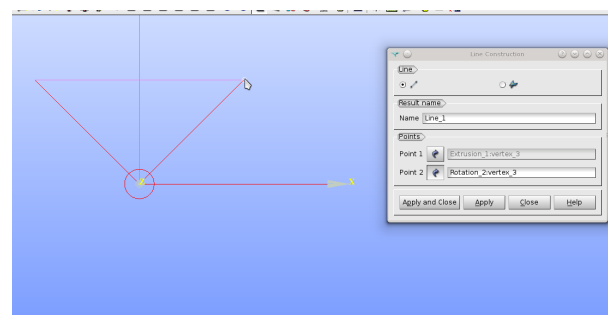
а)



б)



в)



г)

Рис. 1.4.4 Будування відрізків: поворот вектора ОУ (а), витягування точки вздовж вектора (б), поворот відрізка (в), з'єднання кінцевих точок лінією (г)

Відрізок, що зображено на рис. 1.4.4, б), будується методом підвищення розмірності, який можна знайти в меню на панелі управління:

**New Entity → Generation → Extrusion**



Операція Extrusion витягує об'єкта в заданому напрямку, підвищуючи розмірність на один. В нашому випадку точку О розмірності 0D подовжено вздовж нового вектора (рис. 1.4.4, а) на довжину R2. В результаті отримано лінію – об'єкт розмірності 1D (рис. 1.4.4, б).

Відрізок на рис. 1.4.4, в) побудовано аналогічно повороту вектора ОУ на рис 1.4.4, а) за допомогою операції повороту відрізка, що зображено на рис. 1.4.4, б) навколо ОZ на  $-90^\circ$ .

Відрізок на рис. 1.4.4, г) побудовано шляхом з'єднання двох точок на краях відрізків, що зображені на рис. 1.4.4, б) і в). Ці точки необхідно попередньо виділити на вказаних лініях. Конструктор побудови точок і ліній можна знайти в списку базових примітивів на панелі управління:



Порівнюючи побудовані елементи і структуру блоків навколо циліндру на рис. 1.4.1, можна побачити зайві частини ліній. Пакет SALOME дозволяє відокремити необхідні елементи шляхом перерізання одних ліній іншими в меню на панелі управління:

### Operations → Partition.

В якості об'єкта, а також і інструмента розрізання необхідно обрати три елемента: коло Circle\_1, відрізки Extrusion\_1 та Rotation\_2 (рис. 1.4.5).

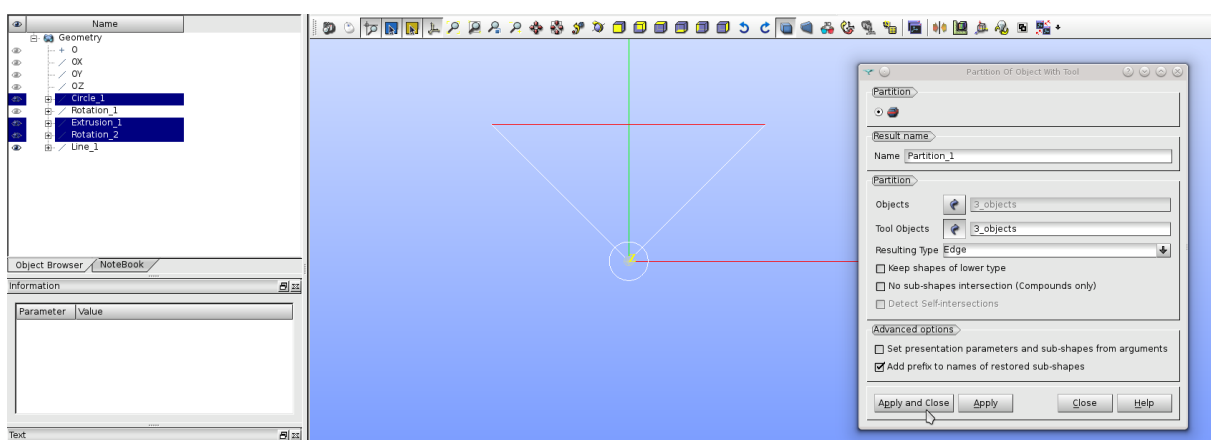


Рис. 1.4.5 Перерізання об'єктів

Після розрізання необхідно застосувати «вибух»:

### New Entity → Explode

Тепер всі розрізані частини доступні окремо для подальшої роботи з геометрією розрахункової області поставленої задачі. Необхідно їх виділити за допомогою команди **Sub-Shapes Selection** на панелі управління, як показано на рис. 1.4.6 білим кольором. Тип під-об'єкту вказати edge, тобто ребро.



Наступним етапом формується двовимірна грань з замкненого контуру (рис. 1.4.7). Для цього використовується команда на панелі управління:

**New Entity → Build → Face**

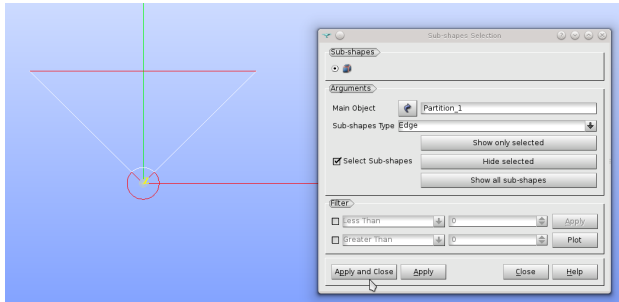
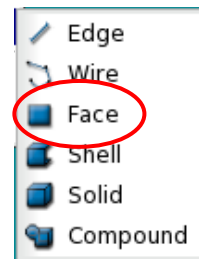


Рис. 1.4.6 Формування ребер з розрізаних частин об'єктів

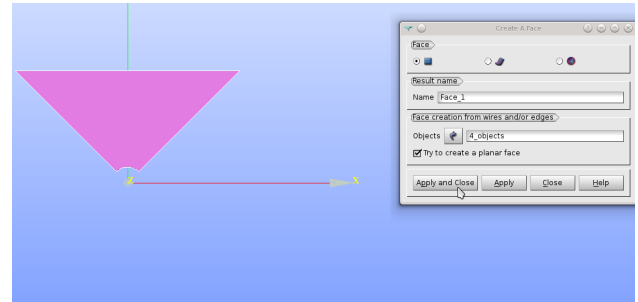


Рис. 1.4.7 Формування 2D поверхні із замкненого контуру

Згідно до постановки задачі необхідно зробити ще три копії грані, що зображена на рис. 1.4.7. Для цього рекомендується використати процедуру багатократного повороту навколо вісі OZ:

**Operations → Transformation → Multi-Rotation**

Результат чотирикратного повороту зображено на рис. 1.4.8.

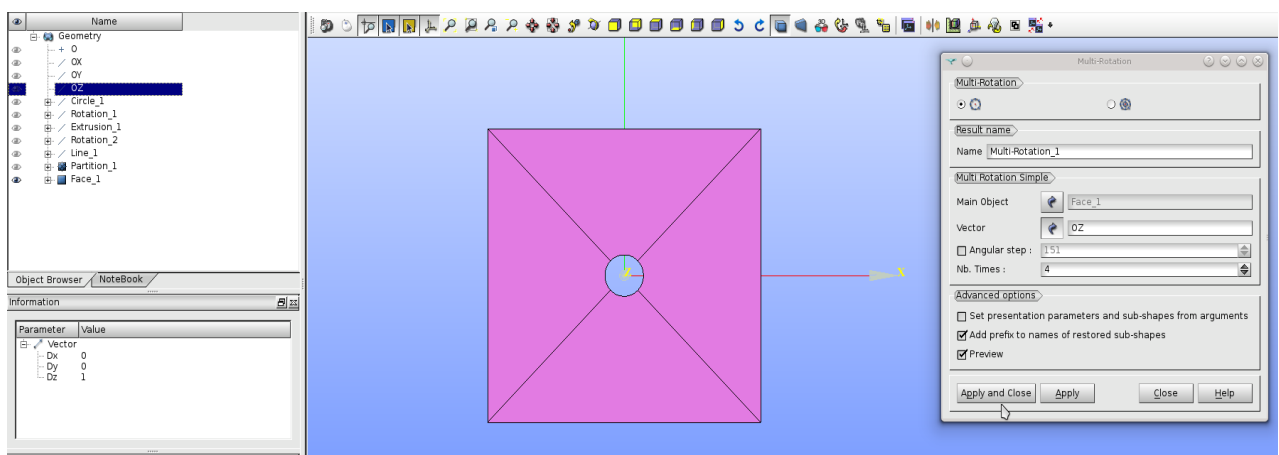


Рис. 1.3.8 Чотирикратний поворот поверхні

Далі необхідно створити об'ємні блоки з побудованих поверхонь методом підвищення розмірності:

### **New Entity → Generation → Extrusion**

На рис. 1.4.9, а) представлено будування об'ємної фігури шляхом витягування двовимірної поверхні Multi-Rotation\_1 в напрямі вісі OZ на довжину z з таблиці параметрів на рис. 1.4.2. Об'ємну фігуру-блок в сліді за обтічним тілом, що зображена на рис. 1.4.9, б), можна «витягнути» з відокремленої бокової поверхні в напрямку OX на параметр L.

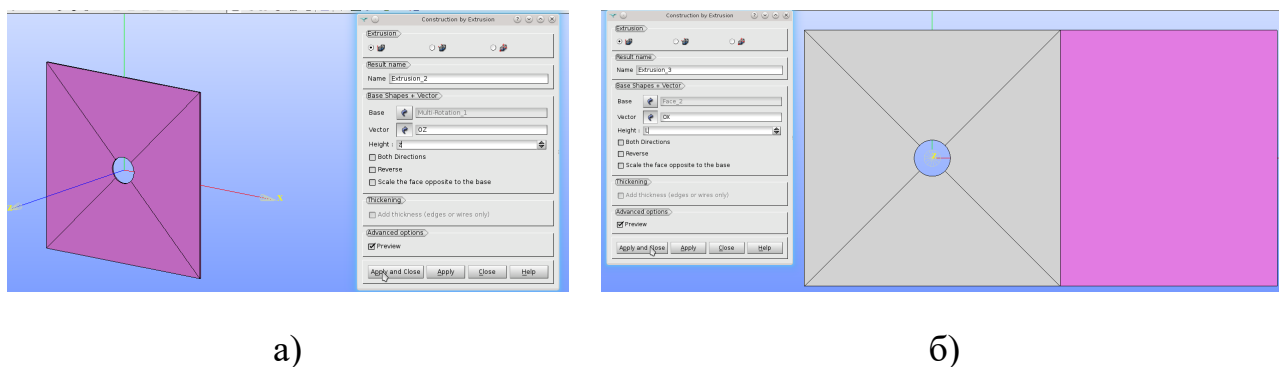
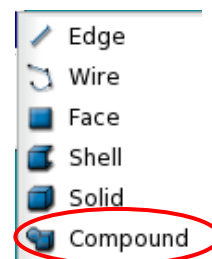


Рис. 1.4.9 Будування 3D об'єму з 2D поверхні: підвищення розмірності фігури в напрямку OZ (а) та в напрямку OX в сліді за обтічним тілом

Об'єднати об'ємні фігури, що представлені на рис. 1.4.9, б), необхідно за допомогою команди на панелі управління:

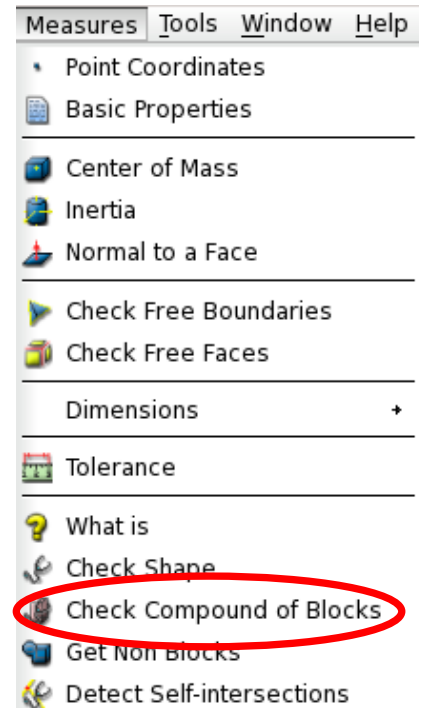
### **New Entity → Build → Compound**



Перед тим, як будувати розрахункову сітку, необхідно перевірити побудовану геометрію. Засоби Measure на панелі управління пакету SALOME включають в себе наступні наприклад:

- Point Coordinates – координати точки;
- Basic Properties – виводить довжину, площу та об'єм об'єкта;
- Center of Mass – центр маси;
- Inertia – виводить моменти інерції;
- Check Free Boundaries – виводить ребра, що належать тільки одній поверхні;
- Check Free Faces – виводить поверхні, що належать тільки одному об'єму;

- Dimensions – містить команди з виведення найменшої відстані, габаритних розмірів і кута між ребрами;
- Tolerance – виводить точність;
- What is – видає кількість елементів усіх типів (вершин, ребер, об'ємів і т.д.) об'єкта;
- Check Shape – перевіряє топологію об'єкта на наявність помилок (неправильна орієнтація нормалей і т. д.);
- Check Compound of Blocks – перевіряє, чи всі об'єми є гексаедральними блоками;
- Get Non Blocks – виділяє об'єми, які не є гексаедральними/квадратичними;
- Detect Self Intersections – перевіряє геометрію на наявність неправильно пересічних елементів.



Блочно-структурована сітка вимагає, щоб всі блоки, з яких складається геометрія розрахункової області, відповідали умовам регулярності: гексаедральні об'єми (6 поверхонь, 12 граней, 8 вершин). Тому, команда Check Compound of Blocks є необхідною. Крім того, вона перевіряє, чи склеєні поверхні сусідніх блоків, та автоматично виправляє помилки, як показано на рис. 1.4.10. Повторна перевірка блочної геометрії не виявить помилок.

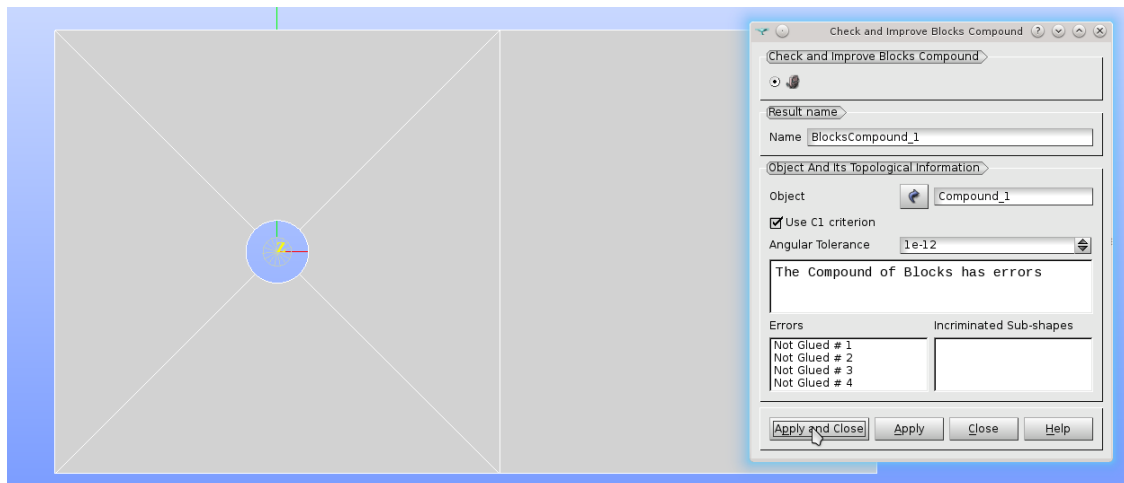


Рис. 1.4.10 Перевірка блочної геометрії, склеювання блоків

### 1.4.2.3 Граничні поверхні

Останнім і дуже важливим етапом побудови геометрії розрахункової області є виділення граничних поверхонь і граней. По-перше, на них будуть задаватися граничні умови в пакеті OpenFOAM. Тому, назви поверхонь в пакеті SALOME мають повністю відповідати назвам поверхонь в задачі OpenFOAM в папці 0. Виділені грані будуть використані при будові підсіток з окремими гіпотезами щільності, напрямку згущення, тощо.

Граничні поверхні (рис. 1.4.11) задаються шляхом виділення частини тривимірної розрахункової області – двовимірної поверхні. Необхідно створити групу об'єктів в меню на панелі управління:

**New Entity → Group → Create Group**

Тип об'єкту (Shape Type) обрати 2D поверхні: inlet, outlet, wall, free. Виділяються фігури простим натисканням і додаванням (кнопка Add).

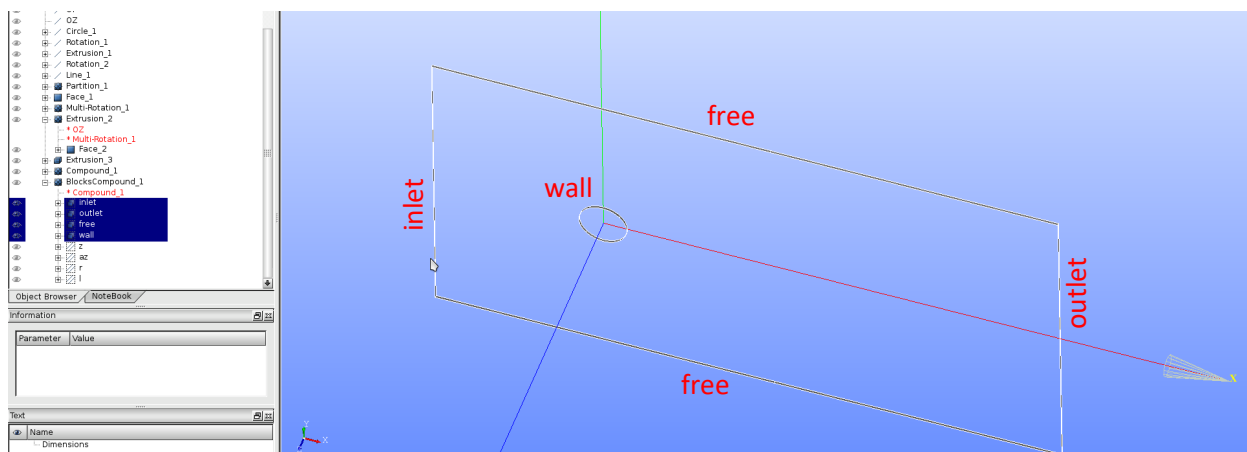


Рис. 1.4.11 Граничні поверхні

Пропонується виділити декілька граней для будування підсіток, що представлені на рис. 1.4.12 білим кольором. Тип об'єкту (Shape Type) обрати 1D грані: z (а), cyl (б), az (в), L (г). Призначення даних граней буде описано в розділі будування підсіток Sub-Mesh.

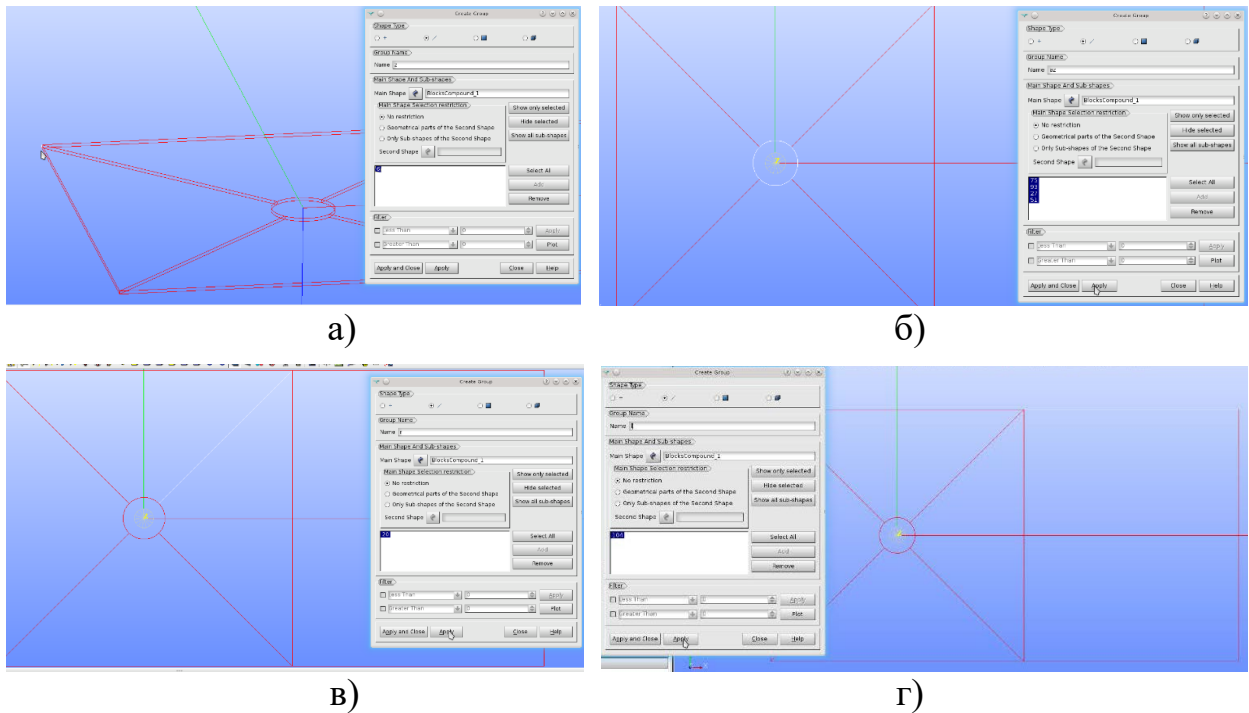


Рис. 1.4.12 Виділені грані для будування підсіток: z (а), cyl (б), az (в), L (г)

## 1.4.3 Побудова розрахункової сітки в пакеті SALOME (модуль Mesh)

### 1.4.3.1 Основна сітка

Створення сітки запускається командою Create Mesh в розділі Mesh панелі управління. Відкривається менеджер створення базової сітки. В якості геометричного об'єкту Geometry обираємо blockCompaund\_1. В меню налаштування обираємо тип гексаедр (рис. 1.4.13, а) і гіпотезу розбиття – число сегментів (рис. 1.4.13, б)

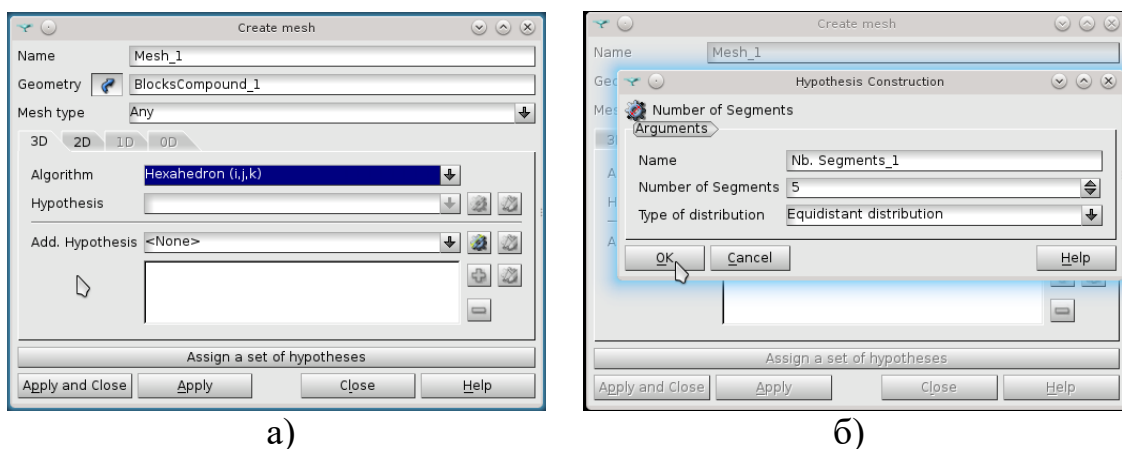


Рис. 1.4.13 Налаштування дискретизації: об'ємів (а), ребер (б)

Командою Compute розраховується сітка з заданими параметрами. На монітор виводиться статистика: форма і кількість елементів сітки (рис.1.3.14).

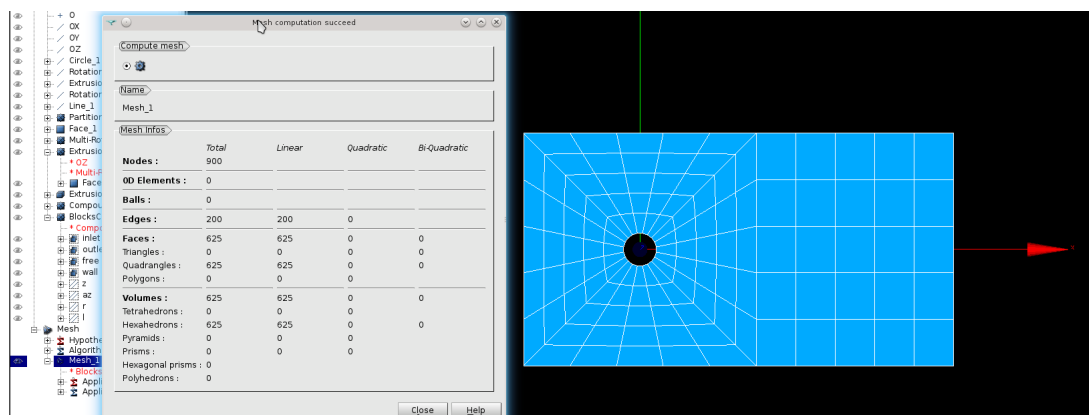


Рис. 1.4.14 Макет основної розрахункової сітки

### 1.4.3.2 Підсітки

В поставленій задачі доречно згустити розрахункову сітку поблизу обтічного циліндру, де спостерігаються найбільші градієнти фізичних величин у зв'язку з опором тертя на твердих поверхнях.

Створення підсітки запускається командою на панелі управління:

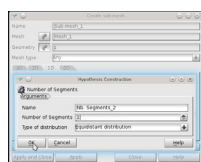
#### Mesh → Create Sub-mesh

При дискретизації розрахункової області, на тій множині, де визначена підсітка використовуються алгоритми, гіпотези і додаткові гіпотези, що відмінні від тих, які задаються для всієї сітки (Mesh).

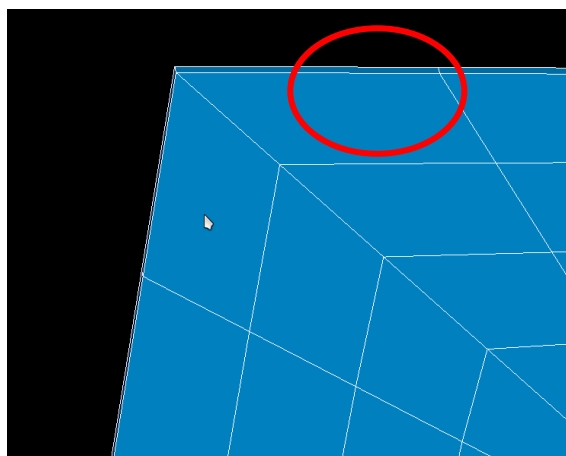
В даній роботі комп'ютерного практикуму пропонується будувати підсітки на основі граней, які було виділено в модулі Geometry (рис. 1.4.12):

- з спеціальна підсітка для 2D задач (рис. 1.4.15).

Зауважимо, що гіпотеза розбиття Nb. Segment побудованої сітки, яка зображена на рис. 1.4.14, застосовується для кожної грані. Таким чином, в напрямку вісі OZ задано теж 5 комірок. Тому, підсітка з одним сегментом у третьому вимірі, яка представлена на рис. 1.4.15, є необхідною умовою. Для 2D задач в третьому вимірі задаються «пусті» граничні умови empty, що виключають розрахунок в даному напрямку.



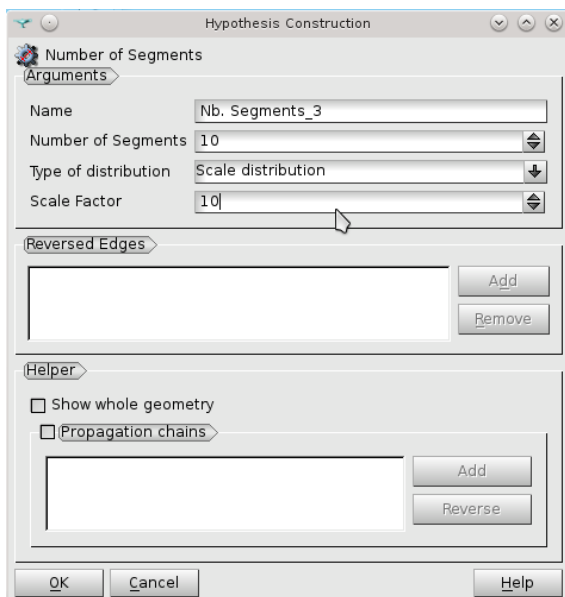
а)



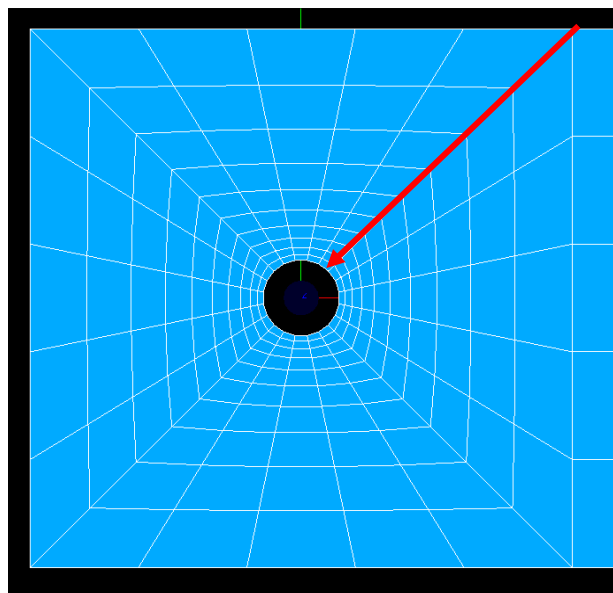
б)

Рис. 1.4.15 Підсітка z: гіпотеза дискретизації (а), перерахована сітка (б)

- az підсітка згущення в напрямку до циліндру (рис. 1.4.16).  
На відміну від рівномірної основної сітки задається масштаб згущення Scale distribution. Scale Factor 10 означає, що найближча до циліндра комірка в 10 разів менше, ніж найвіддаленіша.



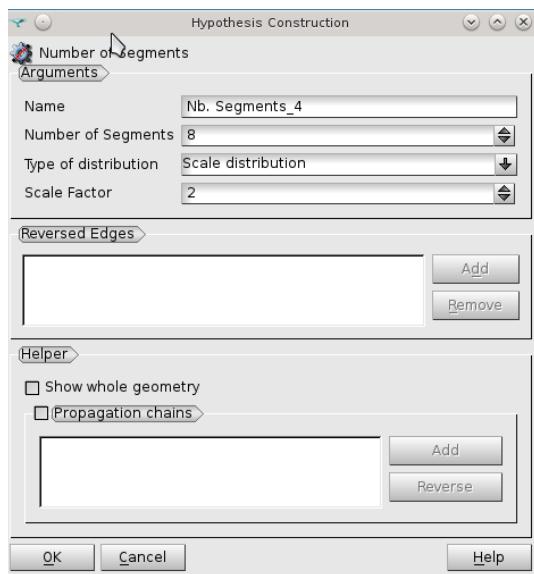
а)



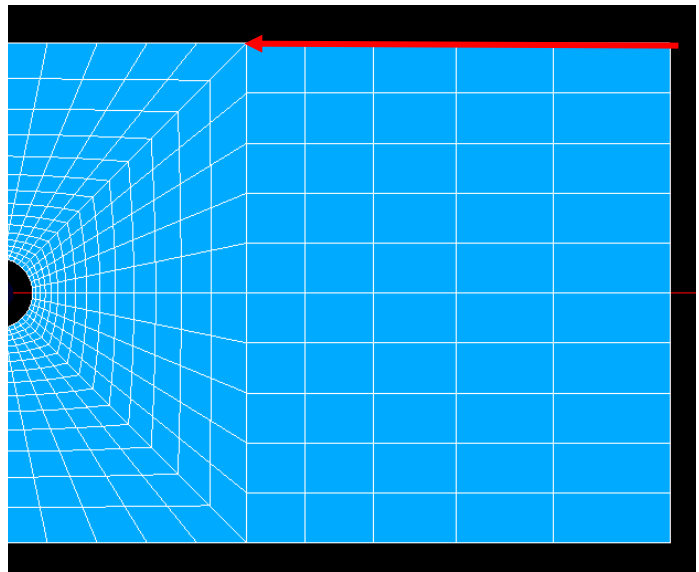
б)

Рис. 1.4.16 Підсітка az: гіпотеза дискретизації (а), перерахована сітка (б)

- L підсітка згущення в сліді за циліндром (рис. 1.4.17).  
Пропонується згущення в сліді за циліндром. В ідеалі, перехід між блоками має бути плавним.



а)



б)

Рис. 1.4.17 Підсітка az: гіпотеза дискретизації (а), перерахована сітка (б)

**Увага!** Налаштовуюючи алгоритми розбиття підсіток, які зображено на рис. 1.4.15-1.4.17, необхідно застосувати додаткову гіпотезу Propagation of 1D Hypothesis on Opposite Edges. Таким чином, в модулі Geometry було обрано тільки одну грань, а гіпотезу розбиття застосовано до всіх протилежних.

### 1.4.3.3 Граничні поверхні

Граничні поверхні, що були виділені в модулі Geometry (рис. 1.4.11) переносяться на побудовану сітку командою

**Mesh → Create Groups From Geometry**

Після імпортування розрахункової сітки з пакету SALOME в OpenFOAM невиділені і неназвані поверхні автоматично об'єднуються в одну групу і отримують назву defaultFaces. В нашому випадку це передня і задня поверхня, до яких будуть застосовано пусті граничні умови.



**Увага!** Не забувайте зберігати етапи будови геометрії та сітки в форматі hdf. Готову розрахункову сітку необхідно експортувати в формат unv.

#### **1.4.4 Підготовка задачі в пакеті OpenFOAM**

Створена за допомогою Salome сітка і збережена у форматі UNV, конвертується в формат OpenFOAM командою `ideasUnvToFoam mesh.unv`, де `mesh.unv` - ім'я файлу.

Пропонується за основу взяти задачу:

**`cd $FOAM_RUN/tutorials/incompressible/pimpleFoam/RAS/pitzDaily`**

При конвертації сітки граничним поверхням автоматично надається тип `patch` (Таблиця 1). Необхідно самостійно змінити тип поверхні для обтічного циліндру на `wall`, а для поверхні `defaultFaces` – тип `empty`.

## Завдання

1. Геометрія розрахункової області.
2. Розрахункова сітка.
3. Розрахунок:
  - Експорт сітки в OpenFOAM **ideasUnvToFoam lab4.unv;**
  - Перевірка сітки **checkMesh;**
  - Зміна одиниць виміру (якщо необхідна)  
**transformPoints-scale '(1e-3 1e-3 1e-3)'**
  - Формування файлів задачі (граничні умови, параметри розрахунку);
  - Запуск розрахунку **pimpleFoam;**
  - Візуалізація результатів **paraFoam.**
4. Самостійне завдання:
  - Побудувати розрахункову сітку методом «знизу-вверх», тобто з створення геометричних об'єктів з поетапним збільшенням розмірності. Сітка має бути з достатньою роздільною здатністю і плавними переходами між блоками.
  - Зробити розрахунки в OpenFOAM і візуалізацію результатів в Paraview.

## Робота 1.5. Обробка та візуалізація результатів розрахунку в графічних пакетах.

Лекційний матеріал. Розділ 6, 7

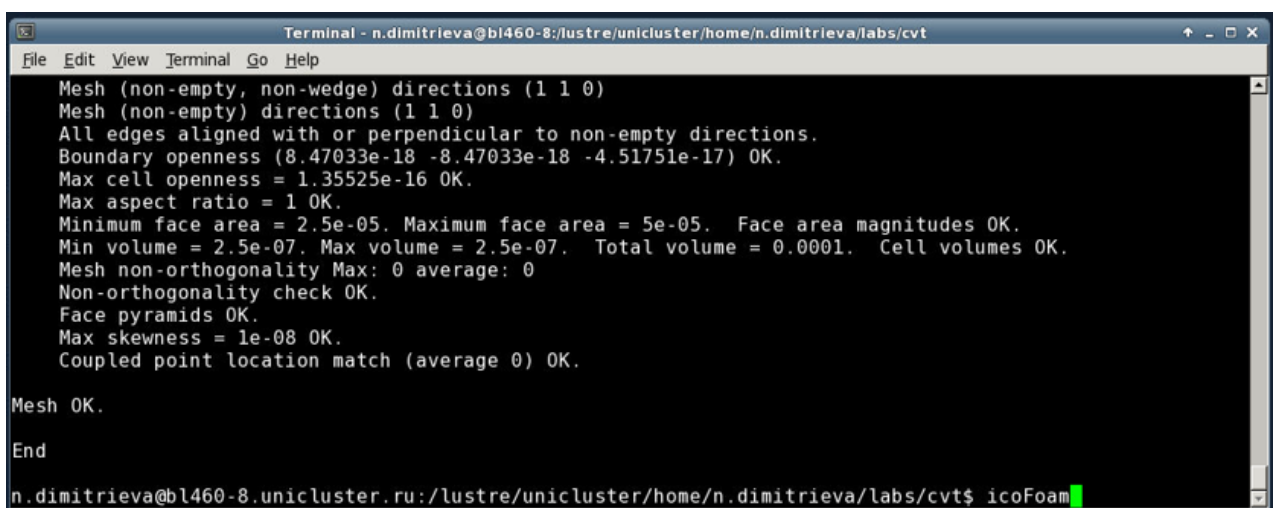
Література: [2, 3, 5-7]

Мета даної роботи комп'ютерного практикуму – навчитися високопродуктивним чисельним експериментам і ефективній обробці та візуалізації результатів обчислень.

### 1.5.1 Чисельний експеримент

#### 1.5.1.1 Запуск задачі в інтерактивному та пакетному режимі

В інтерактивному режимі команди запускаються безпосередньо в терміналі, як продемонстровано на рис. 1.5.1



```
Terminal - n.dimitrieva@bl460-8:/lustre/unicluster/home/n.dimitrieva/labs/cvt
File Edit View Terminal Go Help
Mesh (non-empty, non-wedge) directions (1 1 0)
Mesh (non-empty) directions (1 1 0)
All edges aligned with or perpendicular to non-empty directions.
Boundary openness (8.47033e-18 -8.47033e-18 -4.51751e-17) OK.
Max cell openness = 1.35525e-16 OK.
Max aspect ratio = 1 OK.
Minimum face area = 2.5e-05. Maximum face area = 5e-05. Face area magnitudes OK.
Min volume = 2.5e-07. Max volume = 2.5e-07. Total volume = 0.0001. Cell volumes OK.
Mesh non-orthogonality Max: 0 average: 0
Non-orthogonality check OK.
Face pyramids OK.
Max skewness = 1e-08 OK.
Coupled point location match (average 0) OK.

Mesh OK.

End

n.dimitrieva@bl460-8.unicluster.ru:/lustre/unicluster/home/n.dimitrieva/labs/cvt$ icoFoam
```

Рис. 1.5.1 Запуск задачі в інтерактивному та пакетному режимі

Для запуску задачі в пакетному режимі необхідно створити виконуваний файл, в якому прописується набір команд, які будуть виконуватися за чергою. Структура файлу **\*allRun**:

```
#!/bin/sh
cd ${0%/*} || exit 1 # Run from this directory
# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions
```

```

runApplication checkMesh          // будування сітки
runApplication checkMesh          // перевірка сітки
runApplication decomposePar        // декомпозиція розрахункової області
runApplication mpirun -np 4 pimpleFoam -parallel > log.pimpleFoam      // розрахунки
    в паралельному режимі
runApplication reconstructPar      // збирання результатів паралельних розрахунків
runApplication pimpleFoam > log.pimpleFoam      // запуск розрахунку, якщо
    розпалалювання не передбачене (pimpleFoam – назва чисельної моделі),
    результати прописуються в файл log.pimpleFoam
runApplication foamLog log.pimpleFoam      // обробка файлу log.pimpleFoam
    для подальшого дослідження процесу збіжності результатів обчислень
runApplication -s vorticity postProcess -func vorticity      // утіліта постпроцесінгу, що
    розраховує поле завихреності
#-----

```

### 1.5.1.2 Паралельні розрахунки

Розподілені паралельні обчислення – спосіб розв’язку трудомістких обчислювальних задач з використанням декількох комп’ютерів/процесорів, які об’єднані в паралельну обчислювальну систему. Великі задачі можна розділити на декілька менших, кожна з яких можна розв’язати незалежно від інших.

Даний механізм проведення паралельних розрахунків через декомпозицію розрахункової області вже вбудований в платформу OpenFOAM і не вимагає від користувача доопрацювання програмного коду. Паралельні розрахунки здійснюються в кілька етапів, починаючи з зазначення кількості подобластей, на яке необхідно розділити розрахункову область. Для цього використовуємо утіліту **decomposePar**. Опис методу декомпозиції розрахункової області зазначено у файлі blockMeshDict, що знаходиться в системному каталозі даної задачі:

```

numberOfSubdomains 4; // кількість процесорів
method simple; // метод декомпозиції
simpleCoeffs
{
    n (2 2 1); // Розрахункова область поділяється на 2 в напрямку вісі OX,
    // 2 – Oy та 1 – OZ. Разом 2x2x1=4.
    delta 0.001;
}
hierarchicalCoeffs
{
    n (1 1 1); // За всіма напрямками розрахункова область поділяється на
    // однакові частини.
    delta 0.001;
    order xyz;
}

```

```

manualCoeffs          // ручне налаштування за необхідності.
{
    dataFile           "";
}

```

Необхідно вказати кількість процесорів, метод поділу (ручного або автоматичного) для найбільш ефективного розподілу навантаження між процесорами.

Пакет OpenFOAM пропонує багато методів декомпозиції розрахункової області, зокрема:

- simple – Проста геометрична розбивка, в якій домен розбивається на частини за напрямками, наприклад, 2 частини в напрямку  $x$ , в  $1 - y$  і т.д.
- hierarchical – Ієрархічна геометрична декомпозиція, яка аналогічна простій, за винятком того, що користувач вказує порядок, в якому виконується направлений поділ, наприклад, спочатку в «напрямку  $x$ », потім «напрямок  $y$ » і т. д.
- scotch – Розкладання, яке не вимагає геометричного введення від користувача і намагається мінімізувати кількість меж процесора. Користувач може визначити вагу для розкладання між процесорами через необов'язкове ключове слово `processorWeights`, яке може бути корисно на машинах з різною продуктивністю між процесорами. Існує також необов'язкова стратегія введення ключових слів, яка контролює стратегію декомпозиції через складний рядок, що поставляється в Scotch. Для отримання додаткової інформації див. файл вихідного коду : `$FOAM_SRC/parallel/decompose/scotchDecomp/scotchDecomp.C`
- manual – Ручна декомпозиція, коли користувач прямо вказує розподіл кожного елементу конкретному процесору.

Після виконання команди `decomposePar` в каталозі моделювання (`processor0`, `processor1`, `processor2`, `processor3`) будуть створені чотири нових каталогу, і кожен розрахунок піддомена буде збережений у відповідному каталозі процесорів. Відбувається розподіл між процесорами розрахункової сітки, а також і вихідних даних (крайових умов).

Запускаються обчислення в паралельному режимі командою

**`mpirun -np 4 pimpleFoam -parallel > log.pimpleFoam`**

де `log.pimpleFoam` – це ім'я файлу для збереження даних стану моделювання, замість того, щоб друкувати їх на екрані.

Після завершення обчислень потрібно зібрати підсумковий розв'язок для подальшої обробки і відображення результатів. Утіліта **reconstructPar** збирає всі дані процесорів і створює єдиний комбінований каталог для кожного кроку за часом.

Важливим параметром є прискорення паралельних обчислень “speedup factor”, який розраховується за формулою:

$$S = \frac{T_1}{T_p},$$

де  $S$  – прискорення паралельних обчислень,  $T_1$  – час, що необхідний одному процесору для розв'язку задачі,  $T_p$  – час, що необхідний  $p$  однаковим процесорам для розв'язку задачі. Speedup factor залежить від методу декомпозиції, методу дискретизації, методу розв'язку, тощо. З рис. 1.5.2 видно, що для кожної конкретної задачі існує ліміт, після якого збільшення кількості процесорів неефективне.

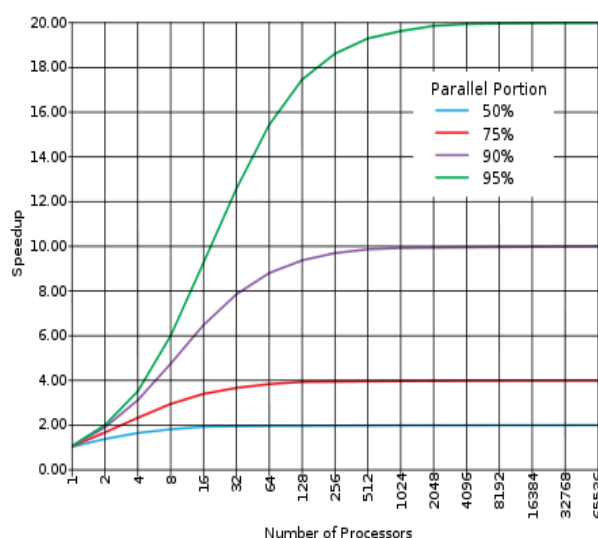


Рис. 1.5.2 Запуск задачі в інтерактивному та пакетному режимі

### 1.5.1.3 Контроль збіжності

Створений в процесі обчислень файл `log.pimpleFoam` з журналом виведення програми може бути в подальшому використаний для побудови графіків збіжності розв'язку. Команда

**foamLog log.pimpleFoam**

генерує файли для всіх можливих параметрів, з файла журналу:

- число куранта, Courant\_0;
- $U_x$  початкові та кінцеві нев'язки  $U_x$ ,  $U_{x\_0}$  и  $U_{xFinalRes\_0}$  та ітерації  $U_{xIters\_0}$  (і еквівалентні дані для  $U_y$ );
- кумулятивні, глобальні та локальні помилки неперервності після кожного з двох рівнянь для тиску  $p$ , contCumulative\_0, contGlobal\_0, contLocal\_0 и contCumulative\_1, contGlobal\_1, contLocal\_1;
- Нев'язки та ітерації з двох рівнянь для  $p$ :  $p\_0$ ,  $p_{FinalRes\_0}$ ,  $p_{Iters\_0}$  и  $p\_1$ ,  $p_{FinalRes\_1}$ ,  $p_{Iters\_1}$ ;
- час виконання розрахунку, executeTime.

Наприклад, хочемо дослідити первинну нев'язку рівняння  $U_x$ , щоб побачити, чи збігається розв'язок до стаціонарного стану. У цьому випадку необхідно побудувати графік залежності з файла logs/ $U_{x\_0}$ , як показано на рис. 1.5.3. Видно, що залишок монотонно падає до тих пір, поки вона не досягне зазначеної точності  $1e-5$ .

Сіткова збіжність також є підтвердженням достовірності отриманих результатів. Комірки розрахункової сітки необхідно зменшувати до тих пір, поки результати обчислень перестануть залежати від їхніх розмірів (рис. 1.5.4).

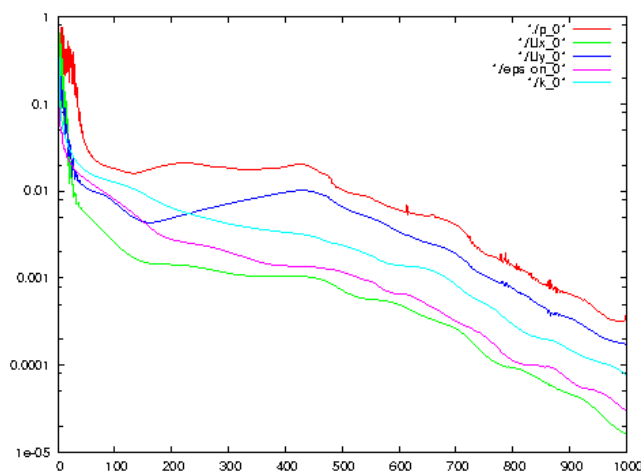


Рис. 1.5.3 Контроль збіжності

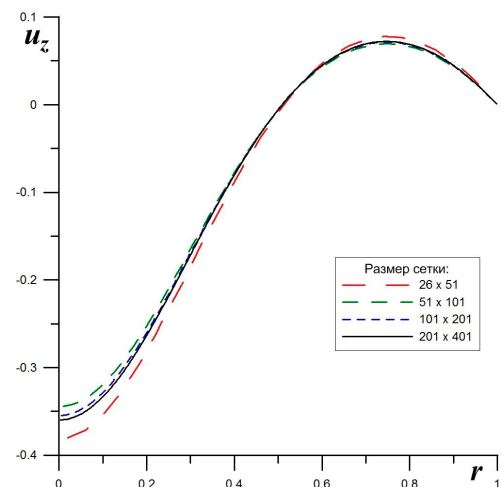


Рис. 1.5.4 Сіткова збіжність

## 1.5.2 Обробка результатів (Postprocessing)

Для повного аналізу фізичного явища, що досліджується, фізичних змінних, які входять в чисельну модель недостатньо. Тому, рекомендується приділити увагу обробці результатів розрахунку (postprocessing).

Пакет OpenFOAM містить великий набір утіліт, які розраховують додаткові локальні та інтегральні фізичні величини, а також перетворюють результати в інші формати для подальшої візуалізації.

Корисні посилання:

- <https://cfd.direct/openfoam/user-guide/post-processing-cli>
- <https://www.openfoam.com/documentation>
- <https://github.com/OpenFOAM/OpenFOAM-5.x/tree/master/etc/caseDicts/postProcessing>

### 1.5.2.1 Обчислення додаткових фізичних змінних (Field calculation)

Пакет OpenFOAM дозволяє розрахувати наступні фізичні величини:

- **vorticity** – поле завихреності, тобто ротор поля швидкостей.
- **wallShearStress** – напруги зсуву в ділянках стінок, виводячи дані у вигляді volVectorField.
- **R** – поле тензора напруг Рейнольдса.
- **components** – скалярні поля компонентів, наприклад,  $U_x$ ,  $U_y$ ,  $U_z$  поля швидкості  $U$ .
- **div** – дивергенція векторного поля.
- **grad** – градієнт скалярного поля.
- **forceCoeffsIncompressible** – коефіцієнти підйомної сили, опору та моменту сил шляхом підсумовування сил на певних ділянках поверхні для випадку, коли чисельна модель нестислива.
- **Q** – другий інваріант тензора градієнта швидкості.
- та інші.

Обчислення додаткових фізичних змінних запускається командою:

**postProcess -func vorticity**

де **vorticity** – назва фізичної величини, поле якої буде обчислено для всіх моментів часу, що містяться в каталозі задачі.

### 1.5.2.2 Контрольні точки (probes)

Користувач надає певні координати точок і список полів, а функція **probes** виписує значення полів саме цих розташувань.



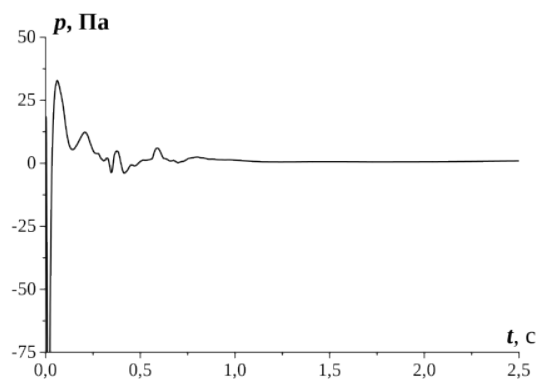


Рис. 1.5.5 Формування поля тиску

За допомогою **probes** проводиться аналіз процесу еволюції фізичних величин в контрольних точках з метою визначення стаціонарності процесу. На рис. 1.5.5 наведено залежність тиску від часу в контрольній точці в якості прикладу такого аналізу. Видно, що після 1 с обчислення не мають сенсу, тому що поле течії цілком сформоване і стає стаціонарним.

Probes визначає найближчі комірки до місць розташування пробної точки і виписує значення фізичних величин комірок. Дані записуються в один файл у форматі час-значення, що підходить для побудови графіку. Користувач може модифікувати probeLocations у в файлі system/controlDict наступним чином:

```
functions
{
    probes
    {
        functionObjectLibs ( "libsampling.so" );
        type                probes;
        outputControl        timeStep;    // Інтервал запису
        outputInterval 5;                // через 5 кроків за часом
        fields ( U p );                // Поля фізичних величин, що виводяться
        probeLocations        // Координати розташування точок
        (
            (0.1 0 0)
            (0.1 -0.5 0)
        );
    }
}
```

Дану інформацію можна занести в окремий файл, а в controlDict вказати посилання:

```
functions
{
    #includeFunc probes
}
```

Після запуску обчислень результати записуються у відповідні файли p та U, що розташовані у postProcessing/probes/0.

### 1.5.2.3 Сили, моменти сил (forces)

Об'єкти функції **forces** та **forceCoeffs** розраховують компоненти повної сили і коефіцієнти сил відповідно: гідродинамічний опір drag, підйомну силу lift, а також момент сили. В файлі controlDict вказується посилання:

```
functions
{
    #includeFunc forceCoeffs
}
```

А відповідний файл містить інформацію, що необхідна для розрахунку коефіцієнтів сил:

```
wall
{
    type forceCoeffs; // вказується тип функції: сили або коефіцієнти
    functionObjectLibs ("libforces.so"); // бібліотека для завантаження
    outputControl timeStep; // Інтервал запису
    outputInterval 5; // через 5 кроків за часом
    patches (torus); // поверхня, на якій обчислюються сили і моменти
    pName p;
    UName U;
    rhoName rhoInf;
    log true;
    rhoInf 1000; // густина рідини
    CofR (0 0 0); // Центр обертання для розрахунку моментів сил
    liftDir (0 1 0); // напрям підйомної сили
    dragDir (1 0 0); // напрям сили опору
    pitchAxis (0 0 1); // вісь обертання для моменту сили
    magUInf 10; // модуль швидкості
    lRef 1.0; // характерна довжина
    Aref 1.0; // характерна площа поперечного перерізу
}
```

Як і probes, інтегральні сили обчислюються на кожному кроці за часом і записуються в папку postProcessing.

### 1.5.2.4 Перетворювання даних після обробки (data converters)

Пакет OpenFOAM постачається з утилітою пост-обробки paraFoam, яка використовує програму візуалізації з відкритим кодом ParaView. Але OpenFOAM записує результати у форматі, який не «розуміють» більшість інших розрахункових і візуалізаційних пакетів.

OpenFOAM містить багато утіліт для попередньої конвертації, зокрема:

- **foamDataToFluent** Перетворює дані OpenFOAM в формат Fluent.
- **foamToEnsight** Перетворює дані OpenFOAM в формат EnSight.
- **foamToEnsightParts** Перетворює дані OpenFOAM в формат Ensight.
- **foamToGMV** Перетворює дані OpenFOAM в формат GMV.
- **foamToTecplot360** Модуль запису двійкових файлів Tecplot.
- **foamToTetDualMesh** Перетворює сітку polyMesh в формат tetDualMesh.
- **foamToVTK** Засіб запису файлів в форматах VTK.

За допомогою утіліти **sample** OpenFOAM надає набір об'єктів функції вибірки даних або через 1D лінію для побудови графіків, або через 2D площину і 3D поверхні для відображення полів фізичних величин. Кожен інструмент вибірки вказується в словнику основних функцій controlDict або в окремих файлах в системному каталозі sampleDict, який містить наступну інформацію:

```
setFormat raw;           // дані для ліній записуються у форматі стовпчиків
surfaceFormat raw;       // дані для поверхонь записуються у форматі стовпчиків
interpolationScheme cellPoint; // лінійна зважена інтерполяція з використанням
                             значень комірок
fields                    // поля фізичних величин, що записуються
(
    P                     // тиск
    vorticity              // завихореність
    U                     // швидкість
);
sets                      // дані для побудови графіків вздовж відрізка через дві точки
(
    lineD1                // назва лінії
    {
        type    uniform;
        axis    distance;
        start   (-0.1 -0.025 0); // координати першої точки
        end     (-0.1 0.025 0);  // координати останньої точки
        nPoints 1000;           // кількість точок
    }
);
surfaces                  // дані для побудови полів на поверхні
(
    constantPlane
    {
        type    plane; // тип площина
        basePoint (0 0 0); // координата базової точки
        normalVector (0 0 1); // напрям вектору нормалі
    }
);
```

Дані можуть бути записані в різних форматах, включаючи добре відомі графічні пакети, такі як Grace/xmgr, gnuplot і jPlot, Origin.

Більш детальний опис можливостей, що надає утіліта **sample** можна знайти за посиланням:

<https://www.openfoam.com/documentation/user-guide/userse21.php>

### 1.5.3 Візуалізація результатів в пакеті Paraview

**ParaView** — відкритий графічний пакет для інтерактивної візуалізації в дослідницьких цілях, що розробляється компанією Kitware. Пакет ParaView надає користувачеві можливості візуалізації і дослідження великих масивів даних для якісного і кількісного аналізу. Пакет ParaView реалізовано на базі бібліотеки Visualization Toolkit (VTK). Пакет ParaView розроблявся для здійснення паралелізму даних на комп'ютерах із загальною, розподіленою пам'яттю і кластерів. Крім того, ParaView може використовуватися і на персональних комп'ютерах.

Корисні посилання:

- Основним джерелом даних має бути в першу чергу сайт:  
<http://www.paraview.org/>  
<https://www.kitware.com/>
- Інструкції:  
<https://www.paraview.org/paraview-guide/>  
[https://www.paraview.org/Wiki/The\\_ParaView\\_Tutorial](https://www.paraview.org/Wiki/The_ParaView_Tutorial)  
<http://paraview.org/Wiki/ParaView>
- Відеоресурси  
<https://www.youtube.com>

В задачах механіки суцільних середовищ наступні можливості пакета можуть бути корисними при аналізі наступних результатів:

- Візуалізація розрахункових сіток (поверхні, сіткові лінії, вершини, об'ємна візуалізація)
- Візуалізація полів скалярних і векторних фізичних величин (тиск, швидкість, температура, зміщення і ін.)
- Побудова графіків, ізо-поверхонь зрізів геометрії площиною або за допомогою заданої функції
- Створення відео, що демонструють розвиток процесу

Деякі можливості візуалізаційного пакета ParaView було описано в методичних вказівках до роботи комп'ютерного практикуму 1 в розділі 1.1.4.

## ParaView. Приладова панель



Деякі основні фільтри (за порядком зліва направо):

- ✓ **калькулятор** — алгебраїчні перетворення над полями;
- ✓ **ізоповерхні** — побудова поверхонь постійного значення;
- ✓ **розсікання** — отримання частини об'єкта, обмеженою певною поверхнею;
- ✓ **переріз** — отримання плоского зрізу об'єкта;
- ✓ **виділення по значенню** — вибір тих комірок, в яких поле задовольняє певній умові;
- ✓ **витяг підмножини** — витягує вибрану підмножину у вигляді самостійного об'єкта;
- ✓ **векторне поле** — візуалізує вибране векторне поле;
- ✓ **лінії течії** — будує траєкторії випадкових частинок під дією векторного поля;
- ✓ **деформація по напрямленню** — деформує об'єкт відповідно до обраного векторного поля;
- ✓ **групування** — групує кілька наборів даних в один
- ✓ **розподіл** — відокремлює один набір даних від загального

## Завдання

1. Запуск задачі в інтерактивному та пакетному режимі.
2. Паралельні розрахунки.
3. Обробка результатів (Postprocessing):
  - Обчислення нових змінних та функцій;
  - Перетворювання даних після обробки утілітою sample.
4. Візуалізація результатів в пакеті Paraview:
  - Візуалізація всіх розрахованих полів скалярних і векторних фізичних величин;
  - Побудова графіків, ізо-поверхонь, зрізів геометрії.

## Робота 1.6. Особливості використання динамічних сіток.

Лекційний матеріал. Розділ 4

Література: [2, 3, 5, 16, 17]

Мета даної роботи комп'ютерного практикуму – навчитися принципам роботи з динамічними сітками в пакеті OpenFOAM.

**Динамічна сітка** – це чисельна модель, що дозволяє розрахунковій сітці змінюватися під час обчислення. Основні властивості динамічних сіток:

- Зміни сітки можуть бути обумовлені встановленим рухом:
  - жорстким рухом тіла або його частини як єдиного цілого;
  - рух частини сітки шляхом згущення і розрідження її комірок;
  - адаптивні сітки – взаємодією характеристик течії з розмірами комірок сітки, наприклад, згущення сітки в високоградієнтних областях.
- Динамічні сітки сумісні з усіма моделями.
- Динамічні сітки можна використовувати з метою:
  - перетворення сітки, щоб врахувати рух тіла;
  - додати/видалити нові клітинки відповідно до критерію;
  - додавання/видалення шарів (нашарування).

Як правило, чисельні моделі з динамічними сітками матимуть в назві термін `DyM`, що включений до назви базової моделі. Наприклад, стандартним базовим вирішувачем може бути `pimpleFoam`, а з рухом сітки – `pimpleDyMFoam`. В нових версіях OpenFOAM немає необхідності в створенні окремої чисельної моделі для динамічних сіток. Рух сітки можна увімкнути, просто додавши файл `dynamicMeshDict`.

### 1.6.1 Постановка задачі

Корисні посилання:

- <https://openfoamwiki.net/index.php/DynamicMeshDict>
- <http://www.wolfdynamics.com/tutorials.html?id=142>
- <https://develop.openfoam.com/Development/openfoam/-/tree/master/tutorials/incompressible/pimpleFoam/laminar/mixerVesselAMI2D>

Чисельна модель    `pimpleFoam`

Задача                `$FOAM_RUN/tutorials/incompressible/pimpleFoam/  
laminar/mixerVesselAMI2D`

В якості прикладу розглянемо нестационарну двовимірну задачу обертального руху.

### 1.6.2 Розрахункова сітка

На рис.1.6.1 продемонстровано геометрію задачі, що вивчається [17]. Розрахункова область складається з двох частин:

- **rotor** внутрішня частина, яка обертається навколо вісі z як єдине ціле;
- **stator** зовнішня нерухома частина.

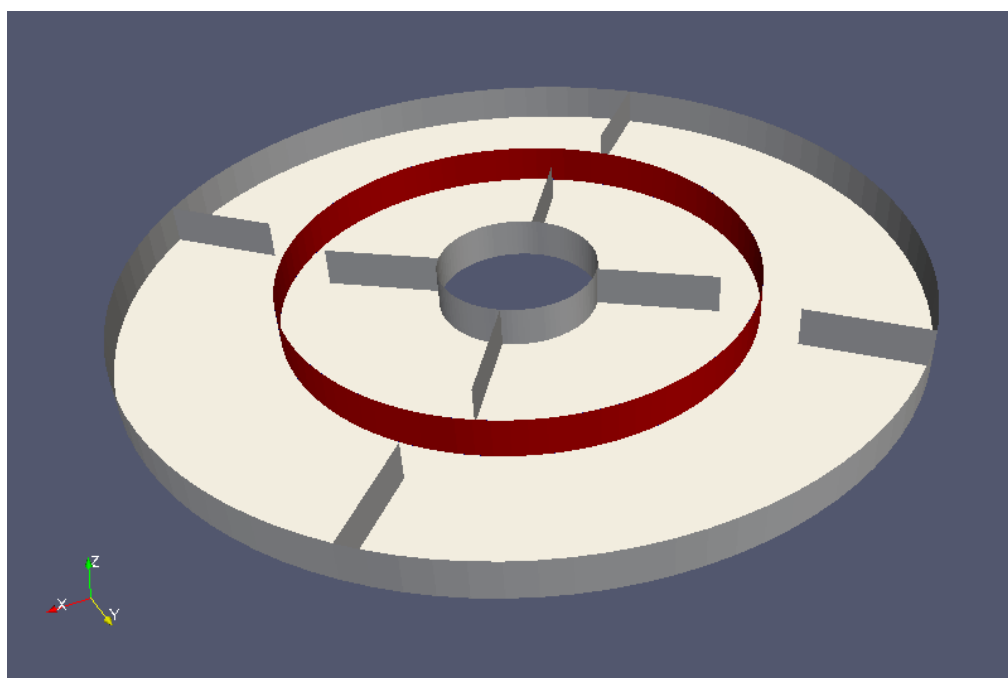


Рис. 1.6.1 Геометрія задачі mixerVesselAMI2D

Для стінок, що показані на рис. 1.6.1 сірим кольором задано стандартний тип wall. Передня front і задня back поверхні – empty, бо для них виключено обчислення і, відповідно, задано тільки одну комірку в напрямку вісі z.

Особливої уваги потребує спільна поверхня між rotor та stator, що позначена на рис. 1.6.1 червоним.

```
AMI1
{
    type      cyclicAMI;
    neighbourPatch AMI2;
    transform  noOrdering;
    ...
}
```

```
AMI2
{
    type      cyclicAMI;
    neighbourPatch AMI1;
    transform  noOrdering;
    ...
}
```

Для них задаються граничні умови cyclicAMI – умова зчеплення між парою поверхонь, які мають спільну межу, але внутрішня конструкція яких може бути різною.

Інформація про рух сітки міститься в файлі constant/dynamicMeshDict:

```
dynamicFvMesh dynamicMotionSolverFvMesh;
motionSolver solidBody; // Жорсткий рух тіла як єдиного цілого
cellZone rotor;        // Частина сітки, що рухається. Щоб повернути весь домен,
                        // необхідно вказати "none" для cellZone, або не включайте цей
                        // запис.
solidBodyMotionFunction rotatingMotion; // Тип руху – обертальний.
origin (0 0 0);         // координати точки, навколо якої обертається сітка.
axis (0 0 1);           // Вісь обертання.
omega 6.5;              // кутова швидкість в рад/с
```

Таким чином, задано обертальний жорсткий рух частини сітки rotor навколо вісі z з кутовою швидкістю 6,5 рад/с.

### 1.6.3 Граничні умови

Згідно до постановки задачі внутрішня поверхня ротора з лопатями має обертатися навколо вісі z, а зовнішня поверхня статора – нерухома. Цей рух реалізується повністю за рахунок обертання частини сітки. А граничні умови для всіх стінок відповідають умовам прилипання. Наприклад, для швидкості U:

```
dimensions [0 1 -1 0 0 0 0];
internalField uniform (0 0 0);

boundaryField
{
    rotor
    {
        type movingWallVelocity;
        value uniform (0 0 0);
    }
    stator
    {
        type movingWallVelocity;
        value uniform (0 0 0);
    }
    front
    {
        type empty;
    }
}
```



```

back
{
    type      empty;
}
AMI1
{
    type      cyclicAMI;
    value     $internalField;
}
AMI2
{
    type      cyclicAMI;
    value     $internalField;
}
}

```

Гранична умова `movingWallVelocity` встановлює бажане значення швидкості при використанні рухомих сіток. Для границь AMI1 та AMI2 задаються граничні умови `cyclicAMI` – умова зчеплення між парою поверхонь, які мають спільну границю, але внутрішня конструкція яких може бути різною. Передня `front` і задня `back` поверхні – пусті граничні умови `empty`.

### Завдання

1. Необхідно самостійно обрати задачу з динамічною сіткою в tutorial із будь-якого розділу (`incompressible`, `compressible`, `multiphase` та ін.). В обраній задачі має бути файл `constant/dynamicMeshDict`.
2. Запустити стандартну задачу, подивитися результати.
3. Потім самостійно змінити рух сітки залежно від заданого руху динамічної сітки (напрямок і швидкість обертання, амплітуду і частоту коливання, тощо) і порівняти отримані результати.

## **КРЕДИТНИЙ МОДУЛЬ**

### **«МОДЕЛЮВАННЯ ФІЗИЧНИХ ПРОЦЕСІВ – 2 ЧАСТИНА»**

#### **ЗАДАЧІ ФІЗИКИ СУЦІЛЬНИХ СЕРЕДОВИЩ**

Кредитний модуль «Моделювання фізичних процесів – 2 частина» розширює знання і навички студентів в комп'ютерному моделюванні фізичних процесів в суцільному середовищі у відкритому програмному забезпеченні SALOME, OpenFOAM, Paraview.

В ході виконання робіт комп'ютерного практикуму студенти отримають навички комп'ютерного моделювання типових задач залежно від фізичних властивостей середовища і режиму течії:

- стаціонарна і нестаціонарна задачі;
- течії нестисливого і стисливого середовища, в тому числі і надзвукові;
- ламінарний і турбулентний режим течії, широкий вибір моделей турбулентності;
- ізотермічні і неізотермічні течії, різні види переносу тепла;
- багатофазні середовища з фазовим переходом і без нього
- та інші моделі.

Теми робіт комп'ютерного практикуму повністю відповідають теоретичному матеріалу лекційних занять. Робочою навчальною програмою даного кредитного модуля передбачено виконання п'яти робіт з різних розділів.

В даному розділі навчального посібника буде представлено короткі рекомендації щодо виконання робіт комп'ютерного практикуму. Також студентам надається право самостійного вибору задач і послідовності їх виконання.

## Робота 2.1. Порівняння моделей турбулентності.

Лекційний матеріал. Розділ 2, 3

Література: [5, 6, 15, 18-24]

### 2.1.1 Постановка задачі

Корисні посилання:

- <https://www.openfoam.com/documentation/guides/latest/doc/guide-turbulence.html>
- <https://cfd.direct/openfoam/user-guide/v6-turbulence/>
- [https://www.cfd.at/sites/default/files/tutorials/2014\\_OFoam\\_Tut\\_Example%20S\\_even.pdf](https://www.cfd.at/sites/default/files/tutorials/2014_OFoam_Tut_Example%20S_even.pdf)
- <https://www.youtube.com/watch?v=ixne2alazKs>
- <https://www.youtube.com/watch?v=fhw8fVnBEwg>

Чисельна модель rhoPimpleFoam, pisoFoam, simpleFoam та ін. (на вибір)

Для виконання даної роботи комп'ютерного практикуму пропонується взяти за основу задачу течії в каналі зі зворотним уступом pitzDaily [15]. Геометрію поставленої задачі представлено на рис. 1.2.1. Але на відмінності від стаціонарної течії нестисливого середовища, що вивчалася в роботі 1.2, можна вибір студента розв'язати задачу турбулентної течії для стисливого газу або для нестисливої рідини, стаціонарну або нестаціонарну.

### 2.1.2 Модель турбулентності

Модель турбулентності задається в файлі constant/turbulenceProperties:

```
simulationType RAS; // Клас моделі турбулентності: RAS, LES або laminar

RAS
{
    RASModel      kEpsilon; // назва моделі турбулентності (в даному випадку k-ε)
    turbulence     on;
    printCoeffs    on;
}
```

Повний список моделей турбулентності, які вбудовано в пакет OpenFOAM можна переглянути в Розділі 7.2 Посібника користувача [5]. Також

список доступних моделей турбулентності, наприклад, для чисельної моделі `pimpleFoam` можна отримати командою:

**`pimpleFoam -listTurbulenceModels`**

### 2.1.3 Візуалізація результатів

На рис. 2.1.1 представлено порівняльний аналіз моделей турбулентності:

- Smagorinsky – модель Смагоринського для підсіткової турбулентності (тип турбулентності LES – Large Eddy Simulation).
- `kEqn` – модель з одним рівнянням вихрової в'язкості (тип LES).
- `kEpsilon` –  $k$ - $\epsilon$  модель турбулентності (тип RAS – Reynolds-averaged simulation)

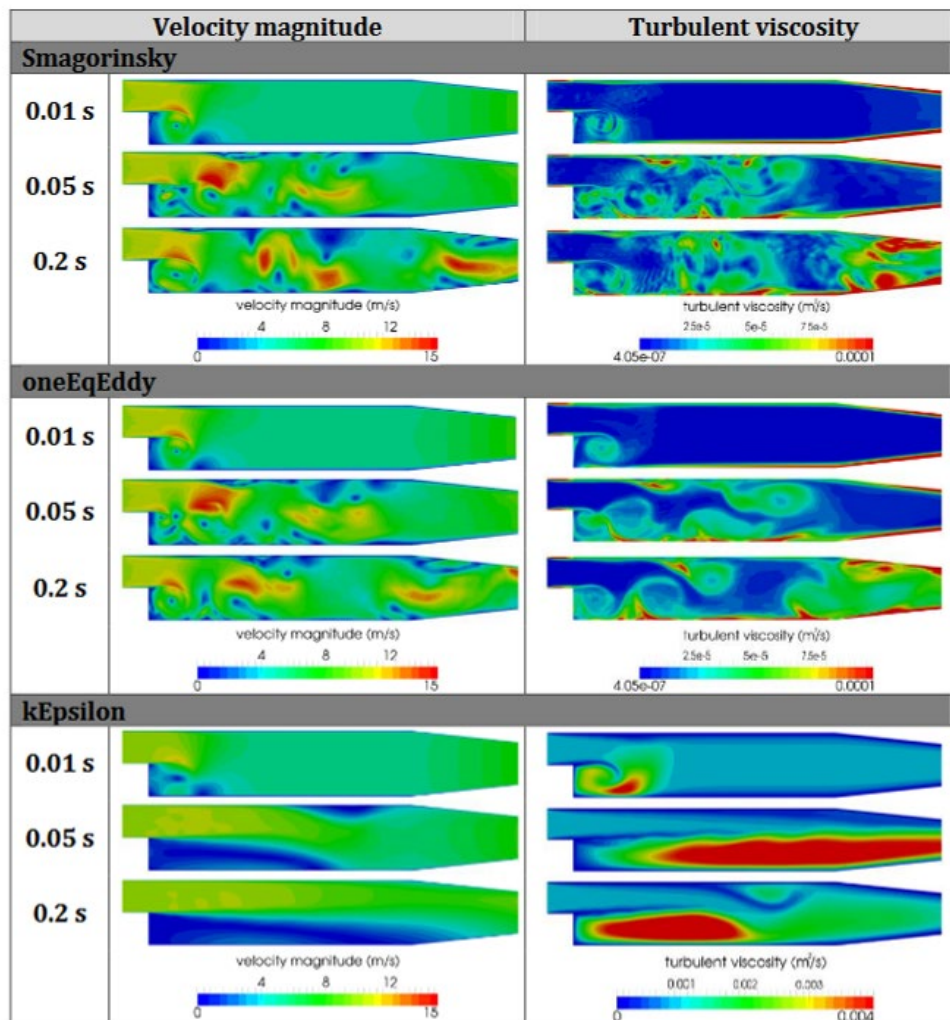


Рис. 2.1.1 Результати обчислень задачі `pitzDaily` з різними моделями турбулентності

### **Завдання**

1. Розв'язати задачу за однакових умов, використовуючи різні моделі турбулентності на ваш вибір.
2. Порівняйте результати.

## Робота 2.2. Розрахунок процесів теплообміну в кімнаті з підігрівом.

Лекційний матеріал. Розділ 2, 5.1

Література: [5, 6, 19, 20, 24-27]

### 2.2.1 Постановка задачі

Корисні посилання:

- <https://develop.openfoam.com/Development/openfoam/-/tree/master/tutorials/heatTransfer/buoyantPimpleFoam/hotRoom>
- <https://www.openfoam.com/documentation/guides/latest/doc/guide-applications-solvers-heat-transfer-buoyantPimpleFoam.html>
- <https://www.youtube.com/watch?v=tZ7Wq740sY8>

Чисельна модель      buoyantPimpleFoam

Задача      \$FOAM\_TUTORIALS/heatTransfer/buoyantPimpleFoam/hotRoom

Досліджується нестационарна тривимірна неізотермічна течія стисливого однорідного ньютонівського середовища в кімнаті з підігрівом. За рахунок різниці температур під дією масових сил (гравітація) виникає конвекція – перенесення теплоти при переміщенні об'ємів середовища (рис. 2.2.1).

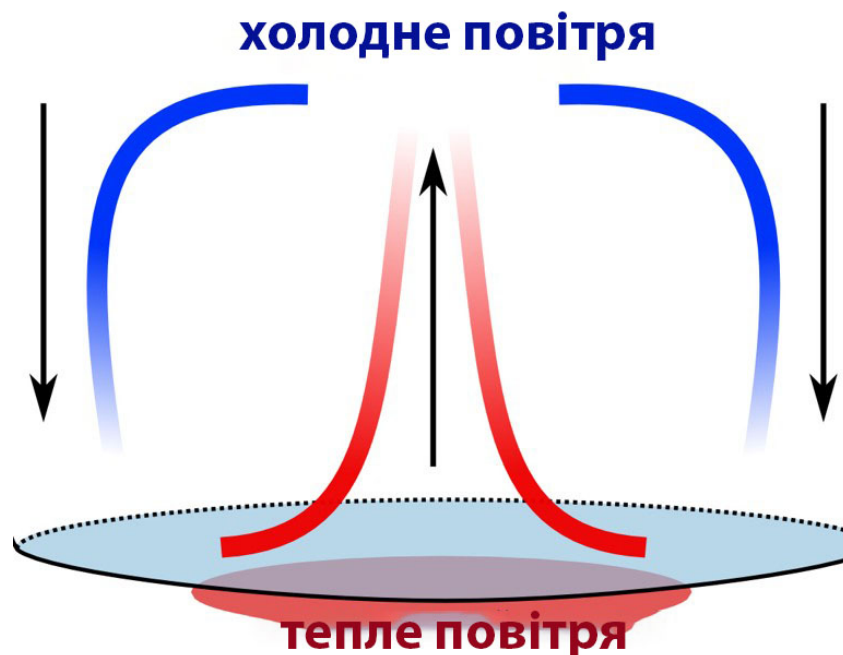


Рис. 2.2.1 Конвекція

До рівнянь Нав'є-Стокса та нерозривності додається стандартне рівняння переносу тепла.

$$\frac{\partial T}{\partial t} + \nabla(T\mathbf{U}) = a\Delta T, \quad (4)$$

де  $\mathbf{U} = \{U_x, U_y, U_z\}$  – вектор швидкості,  $T$  – температура,  $a$  – коефіцієнт температуропровідності.

### 2.2.2 Область підігріву

Розрахункова область представляє собою паралелепіпед з твердими стінками, на яких задано умови прилипання і постійну температуру 300 К. Зону підігріву пропонується задавати за допомогою утіліти препроцесінгу **setFields**. Інформація про зміну значень фізичних величин у визначених зонах міститься в файлі `system/setFieldsDict`:

```
defaultFieldValues
(
    volScalarFieldValue T 300           // поза відокремленої зони температура T=300 К.
);

regions
(
    boxToFace                          // Спеціальні умови будуть застосовані до
    граничної 2D поверхні, яка увійшла до визначеного 3D об'єму.
    {
        box (4.5 -1000 4.5) (5.5 1e-5 5.5); // відокремлюється 3D об'єм – паралелепіпед, що
        визначається через дві точки.

        fieldValues
        (
            volScalarFieldValue T 600     // у відокремленій зоні температура T=600 К.
        );
    }
);
```

Таким чином, на в центрі нижньої стінки розрахункової області задано квадратну зону підігріву з температурою 600 К (рис. 2.2.2).

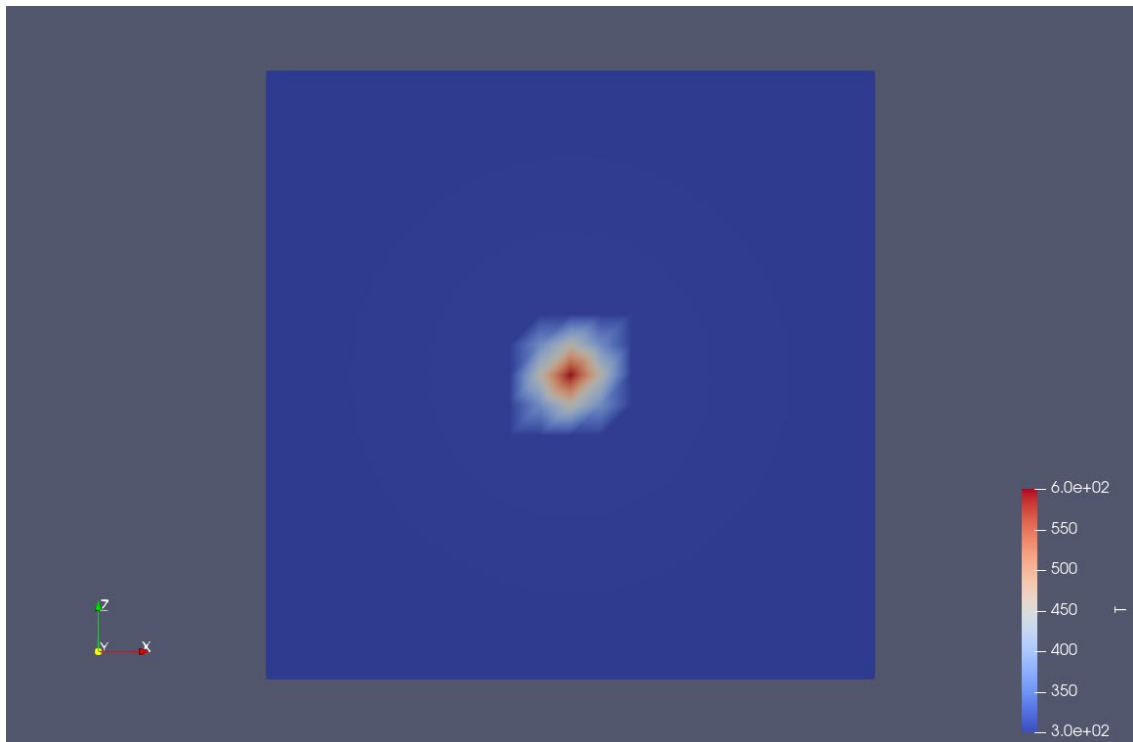


Рис. 2.2.2 Область підігріву

Дана утіліта запускається до початку розрахунків солвера командою:

**setFields**

Основною перевагою утіліти **setFields** є гнучкість та швидкість використання. Тобто, немає необхідності перебудовувати розрахункову сітку при зміні місця розташування зони підігріву.

### 2.2.3 Візуалізація результатів

На рис. 2.2.3 представлено результати обчислень чисельної моделі **buoyantPimpleFoam** на прикладі задачі **hotRoom**. В наслідок різниці температур спостерігається конвективний циркуляційний рух суцільного середовища в замкненому просторі.



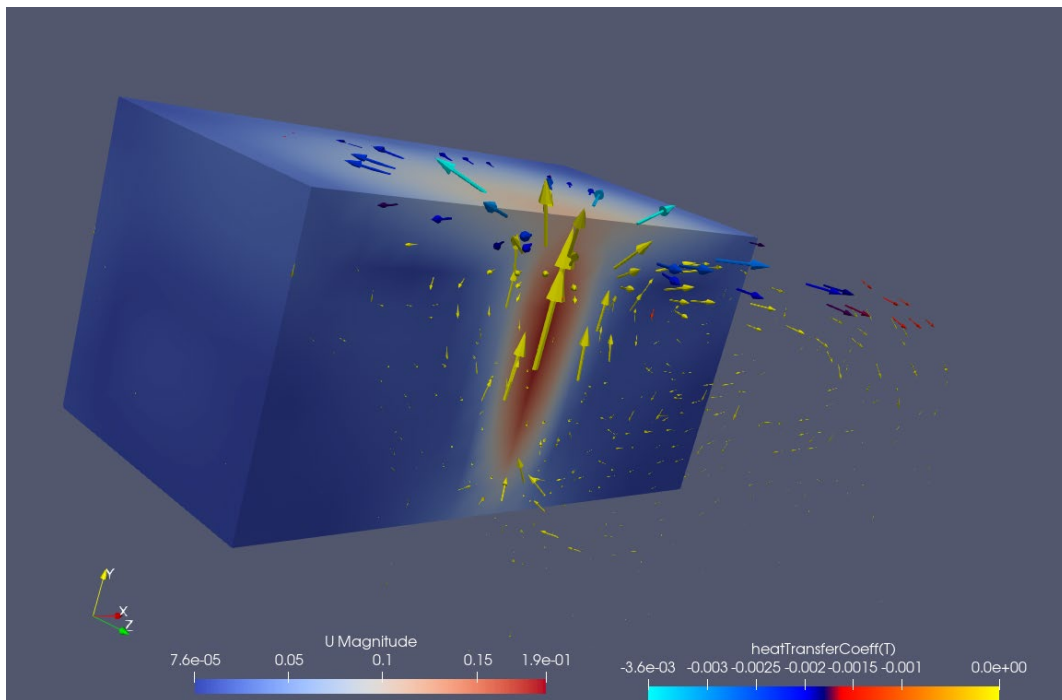


Рис. 2.2.3 Результати обчислень задачі hotRoom

### Завдання

1. Завантажити задачу:  
`cd $FOAM_TUTORIALS/heatTransfer/buoyantPimpleFoam/hotRoom`
2. Розв'язати стандартну задачу. Порядок виконання (команди):
  - Будування сітки **blockMesh;**
  - Перевірка сітки **checkMesh;**
  - Запуск утіліти препроцесінгу **setFields;**
  - Запуск розрахунку **buoyantPimpleFoam;**
  - Візуалізація результатів **paraFoam.**
3. Самостійне завдання:
  - змінити місце розташування нагрівача та розмір: **system/setFieldsDict;**
  - змінити кількість нагрівачів: **system/setFieldsDict;**
  - Дослідити, від яких параметрів залежить інтенсивність теплопереносу та швидкість конвективної течії? Як можемо їх змінити в даній задачі?

## Робота 2.3. Розрахунок двофазного середовища

Лекційний матеріал. Розділ 2, 5.3

Література: [6, 27-29]

### 2.3.1 Постановка задачі

Корисні посилання:

- <https://www.openfoam.com/documentation/tutorial-guide/tutorialse8.php#x14-680004.1>
- <https://develop.openfoam.com/Development/openfoam/-/tree/master/tutorials/multiphase/interFoam/laminar/damBreak/damBreak>
- <https://www.openfoam.com/documentation/guides/latest/doc/guide-applications-solvers-multiphase-interFoam.html>
- <https://cfd.direct/openfoam/user-guide/v8-damBreak/#x7-520002.3>
- <https://www.youtube.com/watch?v=1uhz-ll8Jxc>

Чисельна модель      interFoam

Задача                      \$FOAM\_TUTORIALS/multiphase/interFoam/laminar/  
damBreak/damBreak

Досліджується нестационарна течія двофазового середовища на прикладі спрощеної двовимірної задачі прориву дамби. Особливістю даної постановки задачі полягає у вивченні взаємодії двох нестисливих середових, що не змішуються та розділені вільною поверхнею.

Розрахункова область складається з колонки води в спокої, яка розташована за мембраною з лівого боку ємності. Геометрію і початковий стан представлено на рис. 2.3.1. Червоним кольором зображено рідку фазу води, білим – повітря, сірим – тверду стінку.

В момент часу  $t = 0$  с мембрана видаляється, і стовп води руйнується. Під час обвалення вода стикається з перешкодою на дні резервуара і створює складну структуру потоку.

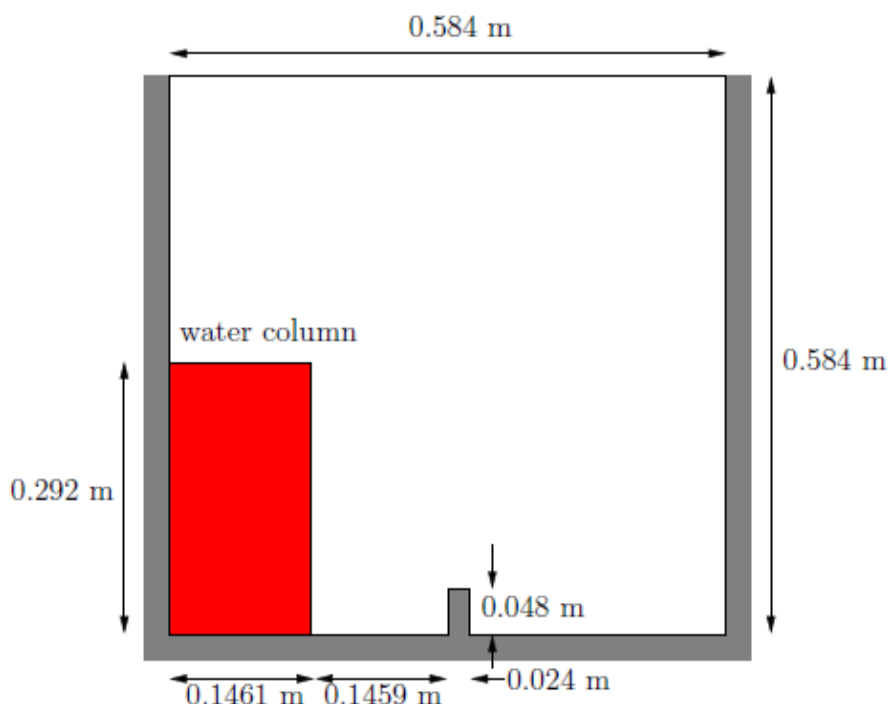


Рис. 2.3.1 Геометрія і початковий стан задачі прориву дамби

### 2.3.2 Метод захвату межі розділу фаз Volume of Fluid

Для розв'язку задачі двофазного потоку використовується метод Volume of Fluid (VOF) – метод, який спирається на визначення індикаторної функції для комірки розрахункової сітки [30]. Вводиться функція  $\alpha$ , яка визначається для точок комірки і приймає значення 1, якщо в точці є рідина. Якщо рідини немає, функція  $\alpha = 0$ . Комірки зі значенням  $\alpha$  від 0 до 1 повинні мати вільну поверхню. Відповідно, градієнти фазової частки зустрічаються лише в зоні зміни фаз.

Часова залежність  $\alpha$  регулюється рівнянням:

$$\frac{\partial \alpha}{\partial t} + \frac{\partial}{\partial x_j} (\alpha u_j) = 0. \quad (5)$$

У цьому рівнянні зазначено, що  $\alpha$  рухається разом з рідиною. Таким чином, метод VOF забезпечує простий та економічний спосіб відстеження вільних меж. В принципі, метод може бути використаний для відстеження поверхонь розриву фізичних властивостей матеріалу.

Одне з найважливіших питань чисельного моделювання потоків з вільною поверхнею за допомогою моделі VOF – це збереження фазової частки. Область поверхні між двома фазами дуже чутлива до роздільної здатності

сітки. У звичайному методі об'єму рідини VOF рівняння переносу для функції індикатора  $\alpha$ , що представляє собою об'ємну частку однієї фази, розв'язується одночасно з рівняннями безперервності та імпульсу [31].

### 2.3.3 Чисельна модель interFoam

Для розв'язування задачі двофазного потоку застосовується стандартна чисельна модель interFoam пакету OpenFOAM для нестационарних ізотермічних двофазних течій, яка відповідає системі рівнянь (1-4). Основні властивості:

- Чисельна модель для двох нестисливих, ізотермічних середовищ без фазового переходу використовує метод захвату межі розділу фаз VOF.
- Імпульс та інші властивості рідини відносяться до «суміші», та розв'язується одне рівняння імпульсу.
- Моделювання турбулентності є загальним, тобто може бути обрано ламінарний режим, RAS чи LES моделі турбулентної течії.

### 2.3.4 Визначення початкового поля

На відміну від попередніх випадків, тепер ми поставимо неоднорідну початкову умову для частки фази води  $\alpha$ . Це досягається запуском утиліти препроцесінгу **setFields**, як і в роботі комп'ютерного практикуму 2.2. Інформація про зміну значень фізичних величин у визначених зонах міститься в файлі system/setFieldsDict:

```
defaultFieldValues
(
    volScalarFieldValue alpha.water 0 // поза відокремленої зони  $\alpha=0$ .
);

regions
(
    boxToCell // Спеціальні умови будуть застосовані до комірок,
               які увійшли до визначеного паралелепіпеди.
    {
        box (0 0 -1) (0.1461 0.292 1); // визначається місце розташування
                                           паралелепіпеди, що визначається через дві точки.
        fieldValues
        (
            volScalarFieldValue alpha.water 1 // у відокремленій зоні  $\alpha=1$ .
        );
    }
);
```

Таким чином, з лівого боку ємності задано об'єм рідини, як зображено на рис. 2.3.1. Утиліта **setFields** визначає початкові умови **boxToCell** до визначеного об'єму на відміну від граничної умови **boxToFace**, що використовувалася в роботі комп'ютерного практикуму 2.2.

Утиліта **setFields** зчитує поле фазової частки  $\alpha$  і після повторного обчислення записує їх у відповідний файл з крайовими умовами **0/alpha.water**. Оскільки файли перевизначаються, рекомендується зробити резервну копію перед виконанням **setFields**. Поле **alpha.water** спочатку рекомендується зберігати лише як резервну копію з назвою **alpha.water.orig**. Перед запуском **setFields** спочатку потрібно скопіювати **alpha.water.orig** в **alpha.water**, наприклад набравши в терміналі команду:

**cp 0/alpha.water.orig 0/alpha.water**

### 2.3.5 Фізичні властивості середовища

Файл **transportProperties** у каталозі **constant** містить властивості матеріалу для кожного середовища, розділеної на два підсловники фаза води та фаза повітря:

```
phases (water air);
water
{
    transportModel Newtonian;           // ньютонівське середовище
    nu      1e-06;                       // коефіцієнт кінематичної в'язкості води
    rho     1000;                         // густина води
}
air
{
    transportModel Newtonian;           // ньютонівське середовище
    nu      1.48e-05;                    // коефіцієнт кінематичної в'язкості повітря
    rho     1;                           // густина повітря
}
sigma     0.07;                         // константа поверхневого натягу між фазами
```

Прискорення вільного падіння є рівномірним по всій розрахунковій області і вказується у файлі з іменем **constant/g**:

```
dimensions [0 1 -2 0 0 0 0];           // розмірність м/с²
value      (0 -9.81 0);                  // означає  $g_y = -9.81 \text{ м}^2/\text{с}^2$ 
```

### 2.3.6 Розрахунковий експеримент

Контроль за ходом розрахунків здійснюється в файлі system/controlDict. Керування кроком у часі є важливою проблемою при відстеженні вільної поверхні, оскільки алгоритм відстеження поверхні значно більш чутливий до числа Куранта  $Co$ , ніж у стандартних розрахунках потоку рідини (див. рівняння (3)). В ідеалі, ми не повинні перевищувати  $Co \approx 0,5$  в області зміни фаз.

### 2.3.8 Візуалізація результатів

На рис. 2.3.2 представлено результати обчислень чисельної моделі interFoam на прикладі нестационарної двофазної задачі прориву дамби через перешкоду. Червоним кольором позначено фазу води, синім – повітря.

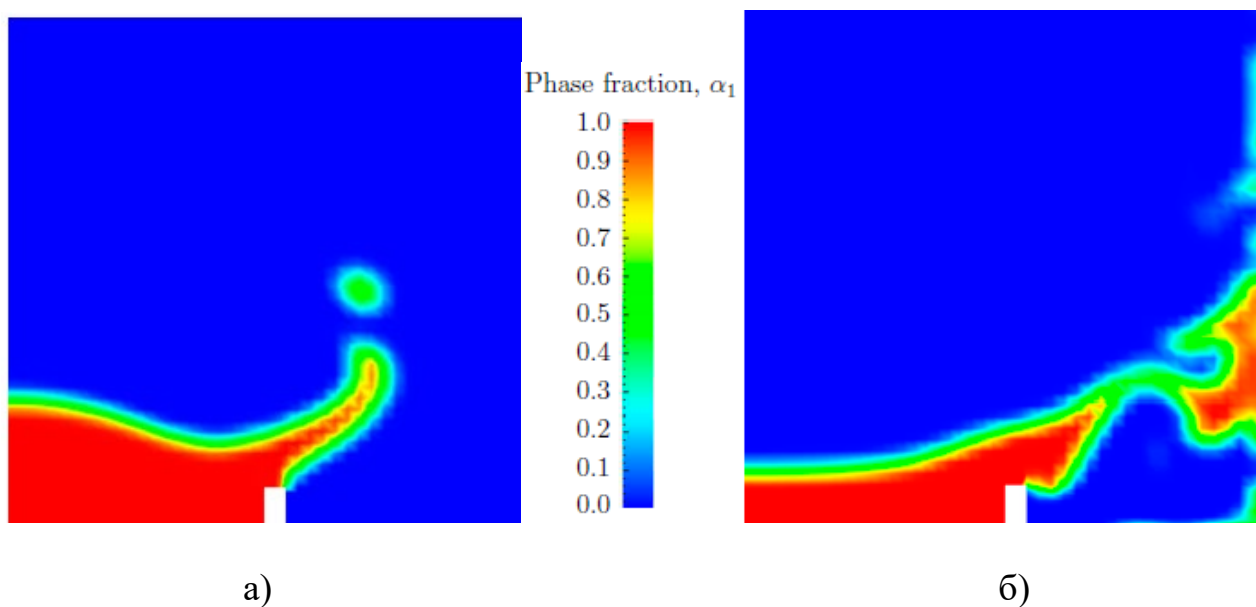


Рис. 2.3.2 Миттєві картини процесу прориву дамби: а)  $t = 0.25$  с; б)  $t = 0.25$  с

## Завдання

1. Завантажити задачу:  
**\$FOAM\_TUTORIALS/multiphase/interFoam/laminar/damBreak/damBreak**
2. Розв'язати стандартну задачу. Порядок виконання (команди):
  - Будування сітки **blockMesh;**
  - Перевірка сітки **checkMesh;**
  - Запуск утіліти препроцесінгу **setFields;**
  - Запуск розрахунку **interFoam;**
  - Візуалізація результатів **paraFoam.**
3. Самостійне завдання:
  - змінити початкові умови для фазової частки рідини (місце розташування, кількість та розмір): **system/setFieldsDict;**

## Робота 2.4. Розрахунок течії стисливого газу.

Лекційний матеріал. Розділ 4

Література: [6, 8, 15, 18, 19, 24, 26, 27]

Мірою впливу стисливості середовища в потоці даної швидкості на його поведінку є характерне число Маха  $M$ :

$$M = \frac{U}{c},$$

що визначається відношенням характерної швидкості руху  $U$  до швидкості звуку  $c$ . Звукові хвилі являють собою малі збурення густини і тиску, швидкість звуку є швидкість поширення цих збурень.

- а) **Дозвукове обтікання ( $M < 1$ )** – Передня частина тіла тисне на частки повітря, що знаходяться попереду нього. З'являються збурення повітря, які поширюються вперед від одних частинок до інших зі швидкістю звуку. Ці збурення випереджають тіло, що рухається. При цьому тіло не зустрічає великого опору.
- б) **Надзвукове обтікання ( $M > 1$ )** – Передня частина тіла буде збурювати частки повітря, породжуючи коливання навколишнього повітря. Але швидкість поширення цих коливань як і раніше дорівнює швидкості звуку. За цих умов коливання повітряного середовища, що виникають при русі тіла, не встигають поширитися попереду тіла. Перед тілом, що рухається з надзвуковою швидкістю, утворюється шар ущільненого повітря.

Важливе значення числа Маха пояснюється тим, що воно визначає, чи перевищує швидкість течії газового середовища (або руху в газі тіла) швидкість звуку чи ні. Надзвукові і дозвукові режими руху мають принципові відмінності; для авіації ця різниця виражається в тому, що при надзвукових режимах виникають вузькі шари швидкого внесення суттєвих змін до параметрів течії (ударні хвилі), що призводять до зростання опору тіл при русі, концентрації теплових потоків біля їхньої поверхні і можливості, тощо.

Тому, при виконанні даної роботи комп'ютерного практикуму пропонується розв'язати характерні задачі для обох режимів течії стисливого середовища: дозвуковий та надзвуковий.



## 2.4.1. Дозвукова течія.

### 2.4.1.1 Постановка задачі

Корисні посилання:

- <https://www.openfoam.com/documentation/tutorial-guide/tutorialse5.php#x10-500003.1>
- <https://develop.openfoam.com/Development/openfoam/-/tree/master/tutorials/compressible/rhoPimpleFoam/LES/pitzDaily>
- <https://www.openfoam.com/documentation/guides/latest/doc/guide-applications-solvers-compressible-rhoPimpleFoam.html>
- <https://github.com/OpenFOAM/OpenFOAM-5.x/tree/master/tutorials/compressible/rhoPimpleFoam/LES/pitzDaily>

Чисельна модель      rhoPimpleFoam

Задача                      \$FOAM\_TUTORIALS/compressible/rhoPimpleFoam/  
LES/pitzDaily

Для виконання даної роботи комп'ютерного практикуму пропонується взяти за основу задачу течії в каналі зі зворотним уступом pitzDaily [15]. Геометрію поставленої задачі представлено на рис. 1.2.1. Але на відмінності від течії нестисливої рідини, що вивчалася в роботі 1.2, необхідно розв'язати задачу течії стисливого газу.

Система рівнянь нестационарної течії однорідного стисливого середовища записується наступним чином:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j} (\rho u_j) &= 0 \\ \frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_j} \left[ \rho u_i u_j + p \delta_{ij} - \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right] &= 0 \\ \frac{\partial \rho e_t}{\partial t} + \frac{\partial}{\partial x_j} \left[ \rho u_j e_t + p u_j - C_p \frac{\mu}{Pr} \frac{\partial T}{\partial x_j} - u_j \left( \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \right] &= 0 \end{aligned} \quad (6)$$

де  $u_i$  – компонента вектора швидкості,  $e_t = e + \frac{u_k u_k}{2}$  – повна енергія,  $p$  – тиск,  $T$  – температура,  $\rho$  – густина,  $\mu$  – коефіцієнт динамічної в'язкості,  $Pr$  – число Прандтля.

### 2.4.1.2 Візуалізація результатів

В результаті обчислень отримаємо нестационарну вихрову течію в плоскому каналі зі зворотнім уступом, що представлена на рис. 2.4.1.

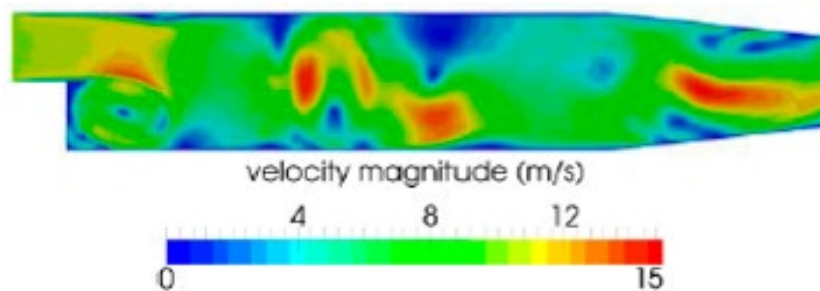


Рис. 2.4.1 Миттєва картина вихрової течії

### Завдання

1. Завантажити задачу:  
**cd \$FOAM\_TUTORIALS/compressible/rhoPimpleFoam/pitzDaily**
2. Порядок виконання (команди):
  - Будування сітки **blockMesh;**
  - Перевірка сітки **checkMesh;**
  - Запуск розрахунку **rhoPimpleFoam;**
  - Візуалізація результатів **paraFoam.**
3. Самостійне завдання:
  - Змінити число Рейнольдса. Порівняти результати.
  - Порівняйте результати розв'язку нестисливої (**pimpleFoam**) та стисливої (**rhoPimpleFoam**) моделі суцільного середовища за однакових чисел Рейнольдса.

## 2.4.2. Надзвукова течія.

### 2.4.2.1 Постановка задачі

Корисні посилання:

- <https://www.openfoam.com/documentation/tutorial-guide/tutorialse6.php>
- <https://develop.openfoam.com/Development/openfoam/-/tree/master/tutorials/compressible/sonicFoam/laminar/forwardStep>
- [https://www.openfoam.com/documentation/guides/latest/api/sonicFoam\\_8C\\_source.html](https://www.openfoam.com/documentation/guides/latest/api/sonicFoam_8C_source.html)
- <https://github.com/OpenFOAM/OpenFOAM-5.x/tree/master/tutorials/compressible/sonicFoam/laminar/forwardStep>
- <https://www.youtube.com/watch?v=z6yWX4dt0iE>

Чисельна модель    sonicFoam

Задача                \$FOAM\_TUTORIALS/compressible/sonicFoam/forwardStep

Досліджується нестационарна двовимірна надзвукова течія зі швидкістю 3 Маха, що задана на вході в прямокутну геометрію перешкодою, яка генерує ударні хвилі (рис. 2.4.2).

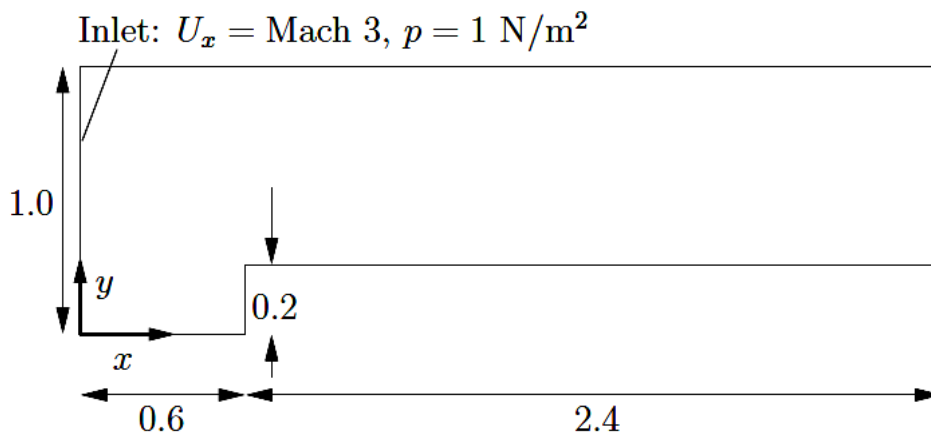


Рис. 2.4.2 Геометрія задачі

Система визначальних рівнянь для надзвукового потоку відповідає моделі ідеального газу:

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) &= 0; \\ p &= \rho R T; \\ \frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) - \nabla \cdot \mu \nabla \mathbf{U} &= -\nabla p; \\ \frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho \mathbf{U} e) - \nabla \cdot \left( \frac{k}{C_v} \right) \nabla e &= p \nabla \cdot \mathbf{U}.\end{aligned}$$

(6)

Тут  $\mathbf{U}$  – вектор швидкості,  $e = C_v T$  – енергія,  $p$  – тиск,  $T$  – температура,  $\rho$  – густина,  $k$  – коефіцієнт теплопровідності, що входить в закон Фур'є  $q = -k \nabla T$ .

#### 1.4.2.2 Крайові умови

Початкові та граничні умови задаються в каталозі 0 для кожної змінної фізичної величини:

- початкові умови:

$$\mathbf{U} = (3 \ 0 \ 0) \text{ м/с}, p = 1 \text{ Па}, T = 1 \text{ К}$$

- граничні умови:

- Inlet: постійна швидкість fixedValue  $\mathbf{U} = 3 \text{ м/с} = 3 \text{ М}$ , тиск  $p = 1 \text{ Па}$  та температура  $T = 1 \text{ К}$ ;
- Outlet: zeroGradient для  $U, p, T$ ;
- No-slip adiabatic wall – умова прилипання на стінці;
- Symmetry plane – умова симетрії.

#### 1.4.2.3 Фізичні властивості середовища

Фізичні і термодинамічні властивості задаються в каталозі constant:

Transport properties:

- Динамічна в'язкість повітря  $\mu = 18,1 \mu\text{Pas}$

Thermodynamic properties

- Питома теплоємність при постійному об'ємі  $C_v = 1,79 \text{ Дж/кгК}$
- Газова постійна  $R = 0,714 \text{ Дж/кгК}$
- Коефіцієнт теплопровідності  $k = 32,3 \text{ мкВт/мК}$

#### 2.4.2.4 Чисельна модель

Використовується чисельна модель (solver) для трансзвукової та надзвукової течії газу **sonicFoam**. Її сконструйовано таким чином, що швидкість звуку газу  $c = \sqrt{\gamma RT} = 1$  м/с, наслідком чого є те, що швидкість еквівалентна числу Маха, наприклад вхідна швидкість  $U_0 = 3$  м/с еквівалентна  $M = 3$ . Цю швидкість обчислення звуку можна перевірити, використовуючи співвідношення для ідеального газу,  $C_p - C_v = R$ , тобто співвідношення:

$$\gamma = \frac{C_p}{C_v} = \frac{R}{C_v} + 1$$

#### 2.4.2.5 Візуалізація результатів

В результаті розрахунків отримаємо прямий від'єднаний стрибок ущільнення (рис. 2.4.3). Напрямок потоку при проходженні через прямий стрибок ущільнення не змінюється. Швидкість за ним стає дозвуковою, що видно на рис. 2.4.3, б). Прямий стрибок ущільнення є межею між надзвуковою і дозвуковою частиною потоку (див. лекційний матеріал).

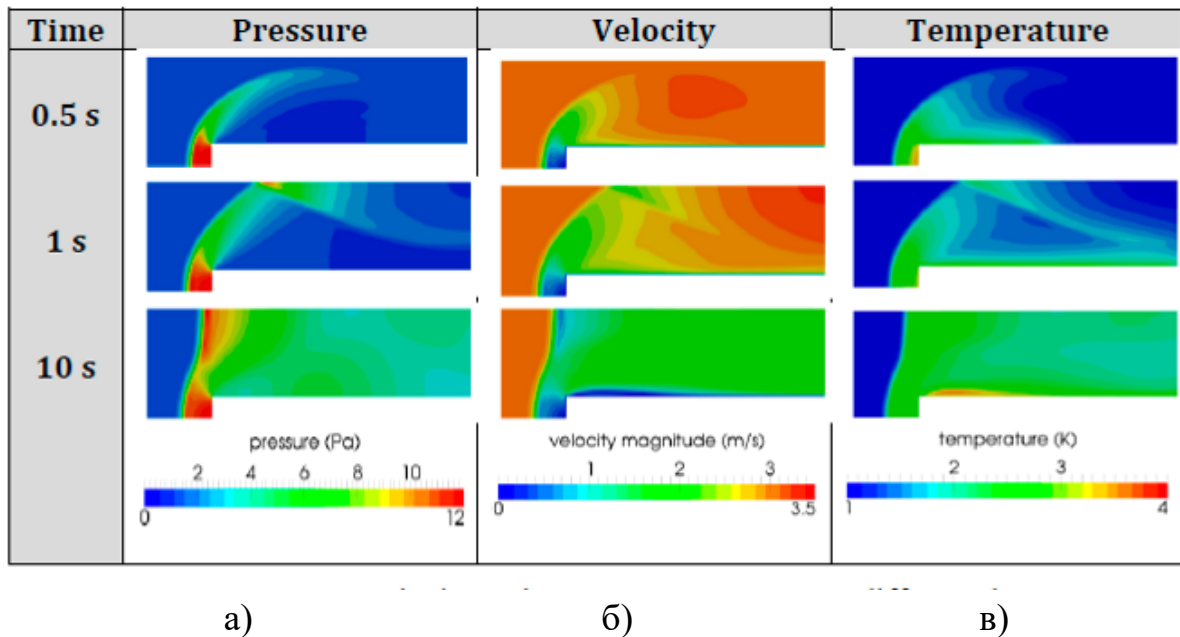


Рис. 2.4.3 Формування надзвукової течії: а) тиск, б) швидкість, в) температура

## Завдання

1. Завантажити задачу:  
**cd \$FOAM\_TUTORIALS/compressible/sonicFoam/forwardStep**
2. Порядок виконання (команди):
  - Будування сітки **blockMesh;**
  - Перевірка сітки **checkMesh;**
  - Запуск розрахунку **sonicFoam;**
  - Візуалізація результатів **paraFoam.**
3. Самостійне завдання:
  - Змінити геометрію таким чином, щоб прямий від'єднаний стрибок ущільнення перетворився в косий приєднаний.
  - Проаналізуйте результати та опишіть відмінності між прямим та косим стрибками ущільнення.

## Робота 2.5. Розрахунок течій в каналах з заданим градієнтом (тепловий, дифузійний або МГД).

Лекційний матеріал. Розділ 5

Література: [19, 20, 24-27, 32]

### 2.5.1 Постановка задачі

Корисні посилання:

- <https://develop.openfoam.com/Development/openfoam/-/tree/master/tutorials/heatTransfer/buoyantBoussinesqPimpleFoam/BenardCells>
- <https://github.com/OpenFOAM/OpenFOAM-5.x/tree/master/tutorials/heatTransfer/buoyantBoussinesqPimpleFoam/BernardCells>
- <https://curiosityfluids.com/2017/06/12/rayleigh-benard-convection-using-buoyantboussinesqpimplefoam/>
- [https://www.youtube.com/watch?v=ItNO\\_nVW1Ps](https://www.youtube.com/watch?v=ItNO_nVW1Ps)

Чисельна модель    buoyantBoussinesqPimpleFoam

Задача                \$FOAM\_TUTORIALS/heatTransfer/  
                         buoyantBoussinesqPimpleFoam/BernardCells

Для виконання даної роботи комп'ютерного практикуму пропонується розв'язати задачу теплообміну при природній конвекції в обмеженому просторі [32].

У горизонтальному шарі суцільного середовища в спокої, який укладений між двома твердими стінками з температурами  $T_{w1} > T_{w2}$  (рис. 2.5.1), передача теплоти здійснюється за рахунок теплопровідності. При цьому розподіл температури лінійний з постійним градієнтом  $(T_{w1} - T_{w2})/h$ , де  $h$  – висота шару.

### 2.5.2 Результати

У випадку підігріву зверху рівновага рідини стійка при будь-яких значеннях градієнта температури (рис. 2.5.1, а). У випадку підігріву знизу більш нагріті шари з меншою густиною виявляються нижчими менш нагрітих шарів, і при градієнті температури, що перевищує критичне значення, виникає вільний конвекційний циркуляційний рух (рис. 2.5.1, б).

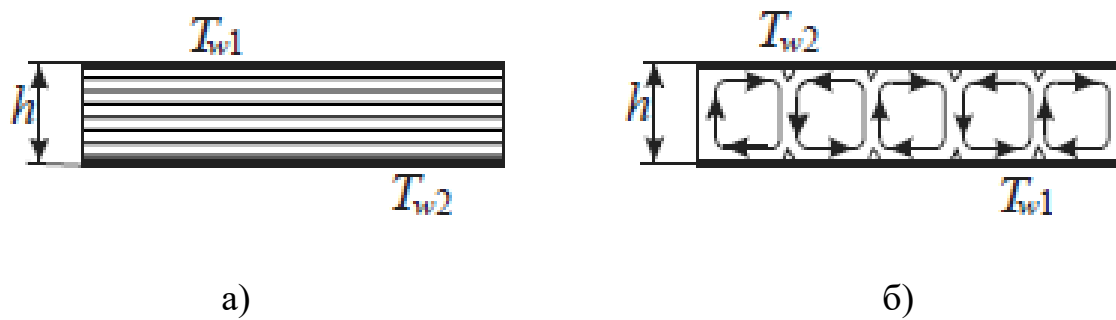


Рис. 2.5.1 Горизонтальний шар рідини: а) – стійка рівновага при підігріві зверху, б) – виникнення конвективних комірок при підігріві знизу

Якісний характер конвекційної течії в вертикальному шарі рідини, що знаходиться між двома стінками з різними температурами, показано на рис. 2.5.2). У цьому випадку структура течії залежить від відношення висоти шару  $H$  і відстані між стінками  $\delta$ . У випадку «широкого» шару закономірності теплообміну та руху рідини наближаються до випадку нескінченного об'єму.

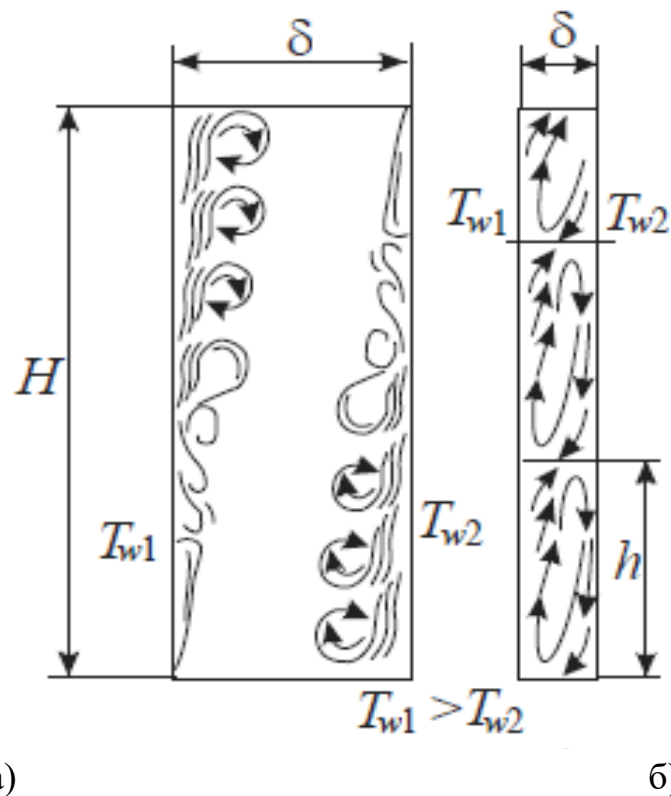


Рис. 2.5.1 Схема природної конвекції в вертикальних шарах рідини: а) – «широкий» шар, б) – «вузький» шар



## Завдання

1. Завантажити задачу:  
**\$FOAM\_TUTORIALS/heatTransfer/buoyantBoussinesqPimpleFoam/  
BernardCells**
2. Порядок виконання (команди):
  - Будування сітки **blockMesh;**
  - Перевірка сітки **checkMesh;**
  - Запуск розрахунку **buoyantBoussinesqPimpleFoam;**
  - Візуалізація результатів **paraFoam.**
3. Самостійне завдання:
  - Дослідити залежність розміру циркуляційних комірок (вихорів) від параметрів задачі: різниця температур, товщина каналу, в'язкість.
  - Дослідити характер конвекційної течії залежно від напрямку дії тили тяжіння.
  - Згідно з теорією подібності процесів переносу можна розглянути задачу циркуляційної течії з розділів теорії дифузії або магнітної гідродинаміки (МГД).

## РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Електронний ресурс – [www.salome-platform.org](http://www.salome-platform.org)
2. Електронний ресурс – <https://www.openfoam.com/>
3. Електронний ресурс – [www.paraview.org](http://www.paraview.org)
4. Комп'ютерне моделювання процесів переносу в суцільному середовищі [Електронний ресурс] : методичні вказівки до виконання курсової роботи з навчальної дисципліни «Моделювання фізичних процесів» для студентів, які навчаються за спеціальністю 105 «Прикладна фізика та наноматеріали» / КПІ ім. Ігоря Сікорського ; уклад. Н. Ф. Димитрієва. – Електронні текстові дані (1 файл: 1,21 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 25 с. – <https://ela.kpi.ua/handle/123456789/33292>
5. OpenFOAM. User Guide // C.J.Greenshields. – The OpenFOAM Foundation, 2020. – 243 p. – <http://foam.sourceforge.net/docs/Guides-a4/OpenFOAMUserGuide-A4.pdf>
6. OpenFOAM. Tutorial Guide. – OpenCFD Limited, 2020. – 104 p. – <https://netix.dl.sourceforge.net/project/openfoam/v2006/TutorialGuide.pdf>
7. Ayachit U. The ParaView Guide. Community Edition / U.Ayachit at al. – Kitware Inc., 2020. – 276 p. <file:///C:/Users/natali/AppData/Local/Temp/ParaViewGuide-5.8.0.pdf>
8. Механіка суцільних середовищ – 1. Механіка суцільних середовищ в інженерних розрахунках [Електронний ресурс] : текст лекцій для студентів спеціальності «Галузеве машинобудування», спеціалізації «Інжиніринг, обладнання та технології виробництв полімерних та будівельних матеріалів і виробів» / КПІ ім. Ігоря Сікорського ; уклад.: О. С. Сахаров, А. Я. Карвацький ; ред. І. О. Мікульонок. – Електронні текстові дані (1 файл: 1,38 МБ). – Київ : КПІ ім. Ігоря Сікорського, 2017. – 233 с. <https://ela.kpi.ua/handle/123456789/19011>
9. Методи обчислень: Частина 1. Чисельні методи алгебри [Електронний ресурс] : навч. посіб. для студ. спеціальності 113 «Прикладна математика», спеціалізації «Наука про дані (Data Science) та математичне моделювання» / КПІ ім. Ігоря Сікорського ; уклад.: В. В. Третиник, Н. Д. Любашенко. – Електронні текстові дані (1 файл: 21,3 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 138 с. – <https://ela.kpi.ua/handle/123456789/28225>
10. Задачин В. М. Чисельні методи : навчальний посібник / В.М.Задачин, І.Г.Конюшенко. – Х. : Вид. ХНЕУ ім. С. Кузнеця, 2014. – 180 с.

11. Попов В. В. Методи обчислень: конспект лекцій для студентів механіко-математичного факультету / В.В.Попов. – Київ: Видавничо-поліграфічний центр "Київський університет", 2012.
12. Uroić T. Implicitly Coupled Pressure–Velocity Solver / T.Uroić, H.Jasak, H.Rusche // In OpenFOAM® Selected Papers of the 11th Workshop, ed. J.M.Nóbrega, H.Jasak, Springer Press, 2019. – pp 103-161. – DOI: [10.1007/978-3-319-60846-4\\_19](https://doi.org/10.1007/978-3-319-60846-4_19)
13. Mathematical Methods in Engineering and Applied Sciences. – Ed. by H. Dutta. – CRC Press, 2020. – 308 p. – ISBN-10: 0367359774, ISBN-13: 978-0367359775
14. Tu J. *Computational Fluid Dynamics: A Practical Approach, 3<sup>d</sup> Edition* / J.Tu, G.H.Yeoh, C.Liu. – Butterworth-Heinemann, 2018. – 498 p. – Paperback ISBN: 9780081011270, eBook ISBN: 9780081012444
15. Pitz R.W., Daily J.W. Combustion in a turbulent mixing layer formed at a rearward facing step. AIAA Journal. 1983. V. 21, no 11. pp. 1565–1570.
16. Електронний ресурс – <https://docs.salome-platform.org/7/gui/GEOM/index.html>
17. Електронний ресурс – <https://www.xsim.info/articles/OpenFOAM/tutorials/incompressible-pimpleDyMFoam-mixerVesselAMI2D.html>
18. Карвацький, А. Я. Механіка суцільних середовищ – 2. Нелінійні задачі механіки суцільних середовищ. Конспект лекцій з навчальної дисципліни [Електронний ресурс] : навчальний посібник для студентів спеціальності 133 «Галузеве машинобудування», спеціалізації «Інжиніринг, комп'ютерне моделювання та проектування обладнання виробництв полімерних і будівельних матеріалів і виробів» / А. Я. Карвацький ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 2,42 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 94 с. – <https://ela.kpi.ua/handle/123456789/23918>
19. Приходько А.А. Компьютерные технологии в аэрогидродинамике и тепломассообмене / А.А.Приходько. – К.: Наукова думка, 2003. – 380 с
20. Халатов, А. А. Основы теории примежового шару [Електронний ресурс] : навчальний посібник для студентів спеціальності 105 «Прикладна фізика та наноматеріали» / А.А.Халатов, Є.В.Мочалін, Н.Ф.Димитрієва ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 2,86 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 191 с. – <https://ela.kpi.ua/handle/123456789/26035>
21. Хинце И.О. Турбулентность. Её механизм и теория / И.О. Хинце. – М.: Физматгиз, 1963. – 680 с.

22. Димитрієва Н. Ф. Моделювання турбулентних течій розведених розчинів полімерів: дис ... канд. фіз.-мат. наук : 01.02.05 / Димитрієва Наталія Федорівна. – Київ, 2011. – 172 с. – <https://ela.kpi.ua/handle/123456789/33718>
23. Rodriguez S. LES and DNS Turbulence Modeling / S.Rodriguez. // In: Applied Computational Fluid Dynamics and Turbulence Modeling. – Springer, Cham. 2019. – pp 197-223. – Print ISBN 978-3-030-28690-3, Online ISBN 978-3-030-28691-0 [https://doi.org/10.1007/978-3-030-28691-0\\_5](https://doi.org/10.1007/978-3-030-28691-0_5)
24. Повх И.Л. Техническая гидромеханика / И.Л.Повх. – Л.: Машиностроение. 1976. – 504 с.
25. Механіка суцільних середовищ [Текст] : підручник / П. Трохимчук ; Східноєвроп. нац. ун-т ім. Лесі Українки. – Луцьк : Вежа-Друк, 2018. – 166 с. – ISBN 978-966-940-114-4
26. Лойцянский Л.Г. Механика жидкости и газа / Л.Г.Лойцянский. – М.: Наука, 1987. – 840 с.
27. Ландау Л. Д. Теоретическая физика, том VI, Гидродинамика / Л.Д.Ландау, Е.М.Лифшиц. М.: Наука, 1986. – 736 с.
28. Нигматулин Р. И. Динамика многофазных сред. Часть I,II. / Р.И. Нигматулин. – М. : Наука, 1987.
29. Multiphase Flow: Theory and Applications. by P.Vorobieff, C.A.Brebbia Ed. – WIT Press, 2018. – 466p. – ISBN-13: 978-1784663117. ISBN-10: 1784663115.
30. Hirt C.W., Nichols B.D. Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries, Journal of Computational Physics, Vol. 39, 1, p. 201-225.
31. Damián S.M. Description and utilization of interFoam multiphase solver/ S.M. Damián // International Center for Computational Methods in Engineering – 2012.
32. Getling, A. V. (1998). Bénard–Rayleigh Convection: Structures and Dynamics. World Scientific. ISBN 978-981-02-2657-2